Manual de Utilização
Série BluePlant

Rev. A 12/2013

Cód. Doc.: MU224000

altus

www.altus.com.br

Conforme o Código de Defesa do Consumidor vigente no Brasil, informamos, a seguir, aos clientes que utilizam nossos produtos aspectos relacionados com a segurança de pessoas e instalações.

Os equipamentos de automação industrial fabricados pela Altus são robustos e confiáveis devido ao rígido controle de qualidade a que são submetidos. No entanto, equipamentos eletrônicos de controle industrial (controladores programáveis, comandos numéricos, etc.) podem causar danos às máquinas ou processos por eles controlados em caso de defeito em suas partes e peças ou de erros de programação ou instalação, podendo inclusive colocar em risco vidas humanas.

O usuário deve analisar as possíveis consequências destes defeitos e providenciar instalações adicionais externas de segurança que, em caso de necessidade, sirvam para preservar a segurança do sistema, principalmente nos casos da instalação inicial e de testes.

Os equipamentos fabricados pela Altus não trazem riscos ambientais diretos, não emitindo nenhum tipo de poluente durante sua utilização. No entanto, no que se refere ao descarte dos equipamentos, é importante salientar que quaisquer componentes eletrônicos incorporados em produtos contêm materiais nocivos à natureza quando descartados de forma inadequada. Recomenda-se, portanto, que quando da inutilização deste tipo de produto, o mesmo seja encaminhado para usinas de reciclagem que deem o devido tratamento para os resíduos.

É imprescindível a leitura completa dos manuais e/ou características técnicas do produto antes da instalação ou utilização do mesmo.

Os exemplos e figuras deste documento são apresentados apenas para fins ilustrativos. Devido às possíveis atualizações e melhorias que os produtos possam incorrer, a Altus não assume a responsabilidade pelo uso destes exemplos e figuras em aplicações reais. Os mesmos devem ser utilizados apenas para auxiliar na familiarização e treinamento do usuário com os produtos e suas características.

A Altus garante os seus equipamentos conforme descrito nas Condições Gerais de Fornecimento, anexada às propostas comerciais.

A Altus garante que seus equipamentos funcionam de acordo com as descrições contidas explicitamente em seus manuais e/ou características técnicas, não garantindo a satisfação de algum tipo particular de aplicação dos equipamentos.

A Altus desconsiderará qualquer outra garantia, direta ou implícita, principalmente quando se tratar de fornecimento de terceiros.

Os pedidos de informações adicionais sobre o fornecimento e/ou características dos equipamentos e serviços Altus devem ser feitos por escrito. A Altus não se responsabiliza por informações fornecidas sobre seus equipamentos sem registro formal.

DIREITOS AUTORAIS

Sumário

1. Suporte Técnico BluePlant

Introdução ao BluePlant

O BluePlant da Altus é a solução definitiva para supervisão e sistemas de aquisição de dados e controle. A reputação da Altus está associada à excelência no fornecimento de sistemas de automação e produtos para controle de processos, como Controladores Programáveis (CPs) e Unidades Terminais Remotas (UTRs), oferecendo desempenho superior, tecnologia no estado da arte e características como redundância, alteração online, troca a quente, alta conectividade e outras funcionalidades avançadas. Esta vasta experiência em sistemas de automação industrial serviu de base para o desenvolvimento deste software SCADA/IHM. A experiência e o portfólio diversificado de produtos garantem à Altus uma posição-chave no fornecimento de soluções completas em automação.

O BluePlant atende requisitos de alto desempenho, recursos de conectividade ampliados, interface gráfica de usuário extremamente rica e poderosa e mecanismos avançados de aquisição de dados em tempo real. A seleção de drivers incorporados no BluePlant, a capacidade de engenharia distribuída, a redundância, o suporte OPC garantem uma experiência nova e única com o BluePlant. Criado no Microsoft Windows Presentation Foundation (WPF), a tecnologia do BluePlant permite obter melhores resultados com as placas gráficas atuais, resultando em aplicações de desempenho excepcional.

O BluePlant inclui também as funcionalidades padrão encontradas nesse tipo de produtos, tais como interação com servidores de banco de dados (SQL, PI, Oracle, Sybase, Informix e outros), barramentos de rede personalizados, módulo de notificação de eventos e servidor de alarme fácil de utilizar, componente de registro e relatório, servidor de histórico avançado, gerenciamento de lógica de negócios, bem como suporte para clientes locais e remotos, executando em computadores, Web, tablets e smartphones.

Figura 1-1. Altus BluePlant

Documentos Relacionados a este Manual

Para obter informações adicionais sobre a Série BluePlant podem ser consultados outros documentos (manuais e características técnicas) além deste. Estes documentos encontram-se disponíveis em sua última revisão em www.altus.com.br.

Cada produto possui um documento denominado Característica Técnica (CT), onde se encontram as características do produto em questão. Adicionalmente o produto pode possuir Manuais de Utilização (os códigos dos manuais são citados na CT).

Aconselha-se o seguinte documento como fonte de informação adicional:

- Características Técnicas BluePlant – CT124000

Inspeção Visual

Antes de proceder à instalação, é recomendável fazer uma inspeção visual cuidadosa dos equipamentos, verificando se não há danos causados pelo transporte. Verifique se todos os componentes de seu pedido estão em perfeito estado. Em caso de defeitos, informe a companhia transportadora e o representante ou distribuidor Altus mais próximo.

É importante registrar o número de série de cada equipamento recebido, bem como as revisões de software, caso existentes. Essas informações serão necessárias caso se necessite contatar o Suporte Técnico da Altus.

Suporte Técnico

Para entrar em contato com o Suporte Técnico da Altus em São Leopoldo, RS, ligue para +55 51 3589-9500. Para conhecer os centros de Suporte Técnico da Altus existentes em outras localidades, consulte nosso site (www.altus.com.br) ou envie um e-mail para altus@altus.com.br.

Se o equipamento já estiver instalado, tenha em mãos as seguintes informações ao solicitar assistência:

- A configuração do sistema instalado (sistema operacional, banco de dados, arquitetura utilizada e atendimento dos requisitos de hardware - computador)
- O modelo de BluePlant utilizado e acessórios adicionais
- O conteúdo do projeto (módulos utilizados), obtido através do programador BluePlant

Mensagens de Advertência Utilizadas neste Manual

Neste manual, as mensagens de advertência apresentarão os seguintes formatos e significados:

PERIGO:
Relatam causas potenciais, que se não observadas, levam a danos à integridade física e saúde, patrimônio, meio ambiente e perda da produção.

CUIDADO:
Relatam detalhes de configuração, aplicação e instalação que devem ser seguidos para evitar condições que possam levar a falha do sistema e suas consequências relacionadas.

ATENÇÃO:
Indicam detalhes importantes de configuração, aplicação ou instalação para obtenção da máxima performance operacional do sistema.

2. Série BluePlant

Características do BluePlant

O BluePlant executa nativamente em máquinas de 64 bits com .NET Framework 4. Há diferentes modelos de produtos para permitir a escolha da melhor solução de acordo com as necessidades. O BluePlant pode atender desde soluções de grande porte até aplicações embarcadas. Os modelos do BluePlant são compatíveis com o legado de computadores de 32 bits. As telas do cliente podem ser executadas em navegadores Web nos computadores com Windows e nos dispositivos móveis baseados em Windows.

	Características
BluePlant Enterprise	Projetado para gerenciamento de plantas de processo, inteligência de negócios (Business Intelligence - BI), painéis em tempo real, SCADA, IHM avançado, controle de processo e otimização. Permite clientes e aquisição de dados distribuídos. O tamanho da aplicação parte de 150 pontos de comunicação
BluePlant Lite	Projetado para painéis, computadores industriais, dispositivos embarcados e sistemas autônomos. Aplicado principalmente em interface de máquinas e pequenos projetos centralizados. O tamanho da aplicação varia de 150 até 1500 pontos de comunicação
BluePlant Express	Projetado somente para avaliação, e não para instalação em campo, limitada em 75 pontos de comunicação e uma hora de execução do Runtime
BluePlant Student	Projetado para escolas e universidades. O tamanho da aplicação varia de 150 até 1500 pontos de comunicação e uma hora de execução do Runtime

Tabela 2-1. Modelos de BluePlant

Para atender as necessidades de integradores de sistemas foram criadas soluções que permitem o desenvolvimento de projetos como no BluePlant Lite ou BluePlant Enterprise. Essas soluções para integradores de sistemas executam o Runtime por apenas uma hora para que possam ser realizados os testes do projeto. Após esse período, é necessário reiniciar o Runtime.

Todas as soluções para integradores de sistemas comunicam plenamente com os controladores programáveis através dos principais drivers de comunicação incorporados nos produtos.

Modelos do BluePlant

O BluePlant é dividido em família/modelos de acordo com as funcionalidades e requisitos de cada aplicação. Desta forma é possível utilizar o modelo de BluePlant adequado ao tamanho da aplicação para otimizar o desempenho do sistema. Deve ser levado em consideração a quantidade total de tags disponíveis, sendo esta quantidade 10 vezes a quantidade de pontos de comunicação. Os pontos de comunicação estão incluídos na quantidade total. Os modelos disponíveis seguem na Tabela 2-1 e na Tabela 2-2 estão as quantidades de pontos de comunicação para cada modelo de BluePlant.

BluePlant Enterprise (Pontos de comunicação)	BluePlant Lite (Pontos de comunicação)	BluePlant Express (Pontos de comunicação)	BluePlant Student (Pontos de comunicação)
-	-	75	-
150	150	-	150
300	300	-	300
500	500	-	500
1500	1500	-	1500
2500	-	-	-
5000	-	-	-
15000	-	-	-
25000	-	-	-
50000	-	-	-
100000	-	-	-
Ultimate	-	-	-

Tabela 2-2. Quantidade de Pontos de Comunicação por Modelo de BluePlant

Produtos Relacionados

A tabela a seguir indica os códigos que devem ser utilizados para adquirir o produto:

	BluePlant Enterprise		BluePlant Lite		BluePlant Express	BluePlant Student
Pontos de comunicação	**Engineering Runtime**	**Runtime**	**Engineering Runtime**	**Runtime**	**Engineering Runtime**	**Engineering Runtime**
75	-	-	-	-	BP6400	-
150	BP1203	BP1103	BP2203	BP2103	-	-
300	BP1205	BP1105	BP2205	BP2105	-	-
500	BP1207	BP1107	BP2207	BP2107	-	-
1.500	BP1209	BP1109	BP2209	BP2109	-	BP4400
2.500	BP1211	BP1111	-	-	-	-
5.000	BP1213	BP1113	-	-	-	-
15.000	BP1215	BP1115	-	-	-	-
25.000	BP1217	BP1117	-	-	-	-
50.000	BP1219	BP1119	-	-	-	-
100.000	BP1221	BP1121	-	-	-	-
Ultimate	BP1299	BP1199	-	-	-	-

Tabela 2-3. Produtos Relacionados por Modelo de BluePlant

Notas:

Ultimate: Aplicações com mais de 100.000 tags devem utilizar essa licença.

BluePlant Enterprise: Esse modelo acompanha um cliente completo para o BluePlant (cliente rich remoto), demais acessórios devem ser requisitados separadamente.

BluePlant Lite: Esse modelo acompanha um cliente completo para o BluePlant (cliente rich remoto) e um cliente web completo para Internet Explorer.

Soluções para Integrador de Sistemas

A tabela abaixo apresenta os códigos de produtos que atendem as necessidades de integradores de sistemas.

Código	Descrição
BP5001	Solução para integradores de sistemas com licenças temporárias – 1 ano
BP5003	Solução para integradores de sistemas com licenças temporárias – 3 anos
BP5010	Solução para integradores de sistemas – BluePlant Lite
BP5020	Solução para integradores de sistemas – BluePlant Lite/Enterprise

Tabela 2-4. Soluções para Integradores de Sistemas

Notas:

BP5001, BP5003: Solução que permite desenvolver projetos tanto no BluePlant Lite como no BluePlant Enterprise, com licenças temporárias válidas por 1 (um) ano (BP5001) e 3 (três) anos (BP5003) a partir da data de gravação.

BP5010: Solução que permite desenvolver projetos para o BluePlant Lite somente, sem limite de validade da licença.

BP5020: Solução que permite desenvolver projetos para o BluePlant Lite e BluePlant Enterprise, sem limite de validade da licença.

Acessórios

A tabela abaixo contém os códigos dos acessórios que podem ser adquiridos. Estes acessórios somente estão disponíveis no modelo BluePlant Enterprise.

Código	Descrição
BP9501	Cliente web de visualização para Internet Explorer
BP9601	Cliente rich completo
BP9701	Cliente web completo para Internet Explorer

Tabela 2-5. Acessórios para BluePlant Enterprise

3. Descrição Técnica do BluePlant

Este capítulo apresenta as características técnicas dos modelos de BluePlant, abordando as partes integrantes do sistema, sua arquitetura e características gerais.

Características Gerais dos Modelos BluePlant

	BluePlant Lite	BluePlant Student	BluePlant Express	BluePlant Enterprise
Limite de execução do Runtime	Não	Sim	Sim	Não
Servidor OPC DA	Sim	Sim	Não	Sim
Linguagem C#	Não	Sim	Sim	Sim
Execução multi-threading de scripts	Não	Não	Não	Sim
Matriz de tags (múltiplas dimensões)	Não	Não	Não	Sim
Tipos do usuário (múltiplos níveis)	Não	Não	Não	Sim
Extensão SDK e integração de toolkits	Não	Não	Não	Sim
Configuração da tabela de histórico	Não	Não	Não	Sim
Servidor para clientes iPad e iPhone	Não	Não	Não	Sim
Clientes rich remotos concorrentes	Não	Sim	Sim	Sim
Clientes web remotos concorrentes (completo e/ou visualização)	Sim, somente um 1 cliente web completo	Sim	Sim	Sim
Redundância de nós de dispositivos	Não	Sim	Sim	Sim
Redundância de servidores	Não	Não	Não	Sim
Relatório através de objetos gráficos	Não	Sim	Sim	Sim
Condições estendidas de alarmes	Não	Sim	Sim	Sim
Controle de versão de projeto	Não	Sim	Sim	Sim
Controle de alterações por objetos	Não	Sim	Sim	Sim
Compressão automática do histórico	Não	Não	Não	Sim
Controle de acesso WPF	Não	Sim	Sim	Sim
Hot start	Não	Não	Não	Sim
Modo teste	Não	Sim	Sim	Sim

Notas:

Limite de execução do Runtime: Execução do Runtime limitada em uma hora. O Runtime pode ser reiniciado.

Linguagem C#: É possível criar scripts utilizando a linguagem C#.

Execução multi-threading de scripts: Essa funcionalidade permite criar scripts e gerar diferentes threads para cada script criado. Quando essa característica está habilitada, a execução das threads é concorrente. Quando essa característica não está habilitada, a execução das threads é sequencial.

Tipos do usuário (múltiplos níveis): É permitido criar novos tipos de dados, sendo possível utilizar até quatro níveis encadeados de tipos de dados.

Extensão SDK e integração de toolkits: É possível criar bibliotecas proprietárias (dlls), com funções específicas e utilizar em projetos.

Clientes rich remotos concorrentes: BluePlant Enterprise deve ser instalado na máquina remota e as licenças devem estar presentes na máquina servidora. O número de clientes rich remotos concorrentes depende do número de licenças adquiridas.

Clientes web remotos concorrentes (completo e/ou visualização): O número de clientes web remotos concorrentes tanto de visualização quanto o completo depende do número de licenças adquiridas, exceto para o BluePlant Lite que permite somente um cliente remoto completo e nenhum cliente remoto de visualização.

Redundância de servidores: Para utilizar essa funcionalidade são necessários dois servidores e duas hardkeys com as suas respectivas licenças. Cada servidor deve ter a sua própria hardkey e então é possível configurar os servidores como um par redundante.

Controle de alterações por objetos: Essa característica permite controlar as alterações feitas em telas, tags, scripts, módulos e outras modificações em objetos no projeto.

Hot start: É possível modificar a aplicação e recarregá-la sem parar o sistema.

Características Gerais Comuns

	BluePlant Lite, BluePlant Express, BluePlant Student e BluePlant Enterprise
Protocolos simultâneos	Todas os modelos do BluePlant possuem pelo menos quatro canais
Cliente OPC	Sim
Abertura de múltiplos projetos	Sim
Linguagem VisualBasic .NET	Sim
Scripts para avaliações de expressões matemáticas.	Sim
Scripts para criar classes .NET e tarefas	Sim
Integração com base de dados SQL	Sim
Historiador e Logger	Sim
Alarme e Proteção	Sim
Editor Gráfico WPF	Sim
Ferramentas de engenharia e depuração	Sim
Editor de relatório	Sim
Controle de alterações por tabelas	Sim
Localization	Sim

Notas:

Protocolos simultâneos: Execução simultânea de drivers de comunicação durante a execução do Runtime.

Controle de alterações por tabelas: O controle de alterações por tabelas informa o que foi modificado, removido ou inserido, mas não informa onde as alterações foram feitas.

Localization: Essa característica traduz textos de telas e alarmes no Runtime.

Características Gerais

Software com Segurança Intrínseca

A característica de Segurança Intrínseca confere segurança e confiabilidade ao produto, uma das bases fundamentais para o desenvolvimento da plataforma BluePlant é que não há nenhum uso de código C ou C++, eliminando totalmente o risco de problemas com ponteiro e/ou exceções de memória. Cada tarefa de execução e processo, seja interna ou criada para ser executada no âmbito do BluePlant, é executada em seu próprio espaço alocado e "protegido" com controle de exceção interno, isolamento de memória, controle multitarefa e sincronização em tempo real. A metodologia de desenvolvimento de software empregada com VBA, VBScript, matemática e lógica proprietária, onde os potenciais problemas podem ser detectados em Runtime, foram eliminados e substituídos por linguagens compiladas do .NET, com validação completa dos scripts antes da execução pelo Runtime com proteção interna que confere desempenho superior, maior estabilidade operacional e segurança.

Mecanismo Gráfico Superior

Os gráficos no BluePlant são do tipo Windows Presentation Foundation (WPF) com sistema interno utilizando XAML. Isso permite a integração com mapas geoespaciais e modelos 3D. Os modelos 3D podem ser apresentados diretamente, bem como vinculados a dados dinâmicos com respostas associadas e comportamentos baseados em eventos e valores em tempo real. Um poderoso editor gráfico WPF está incluído no BluePlant. Os clientes Web dependem de XBAP (aplicativos baseados em navegador e Silverlight) de forma que não há nenhuma exigência para a instalação de qualquer componente externo ActiveX. Além do suporte que permite que páginas da Web sejam apresentadas em smartphones, o BluePlant oferece suporte para gráficos e relatórios em iPad e iPhone da Apple.

Recursos de Manutenção, Teste e Diagnóstico Avançado

O sistema possibilita a troca de versões do projeto, permitindo que as aplicações em modo teste sejam executadas lado a lado, no mesmo servidor, com as aplicações no modo produção para validação e garantia da qualidade, incluindo a análise da utilização da UCP e das estatísticas de comunicação dos módulos do Runtime e redes fornecidas. As opções de hot-standby interna para redundância, locais alternativos de funcionamento e recuperação de desastres também estão incorporadas.

Servidores e Extensões .NET Incorporados

Além dos módulos incorporados para banco de dados em tempo real, SQL externo e conexões ERP, servidor de alarmes e eventos, servidor de históricos e emissão de relatórios, o BluePlant permite acesso completo ao Microsoft .NET Framework, permitindo customização e extensibilidade avançadas, sem o acréscimo de qualquer tipo de aplicação de terceiros ou ferramenta externa.

Além das características indicadas anteriormente, o BluePlant foi projetado a partir de um "campo verde" e foi criado totalmente sem empregar qualquer código herdado. É um aplicativo de código

100% gerenciado que permite utilizar e aproveitar o potencial pleno do Microsoft .NET Framework tanto atualmente, como futuramente. O BluePlant possui uma interface de configuração criada inteiramente a partir do Microsoft Windows Presentation Foundation Graphics (WPF) e suporta integralmente software como serviços (SaaS) implantação combinada com instalações locais, permitindo acessar e colaborar no desenvolvimento de projetos em todo o mundo com apenas um navegador de internet.

Características Inovadoras do Produto

Banco de Dados em Tempo Real (Tags)

O BluePlant suporta os seguintes tipos de pontos de dados em tempo real: Digital (Booleano), Analógico (Inteiro, Double e Decimal), Mensagem de Texto, Contador, Temporizador e variáveis de data e hora. Além disso, podem ser definidos tipos próprios com vários níveis de encadeamento, tags de referência e matrizes tridimensionais.

O banco de dados em tempo real garante, sem a necessidade de qualquer programação adicional, a sincronização de dados entre vários processos no servidor e múltiplas estações cliente. Um vasto conjunto de propriedades internas, tais como qualidade de dados, estampa de tempo, bloqueio de estado e valor bloqueado, permitem e simplificam a criação de aplicações.

	Banco de dados em tempo real (Tags)
Suporte estendido a tipos de tags	Digital, Analógico Inteiro, Analógico Double, Analógico Decimal, Texto, Temporizador, Contador, Date/Time (variáveis de data e hora)
Tag do tipo tabela de dados incorporado	Acesso aos resultados da consulta do conjunto de dados no objeto de tabela de dados .NET padrão
Tags de referência	Utilização de tags de referência para chavear o link do Tag na execução do Runtime
Tag do tipo matriz	Definição de tags do tipo matriz (uma a três dimensões, dependendo da versão do produto)
Tipos e estruturas definidas pelo usuário	Definição de tipos próprios para o banco de dados em tempo real
Propriedades do Tag	Amplo conjunto de propriedades de tag acessíveis na configuração do Runtime

Tabela 3-1. Banco de Dados em Tempo Real (Tags)

Linguagens .NET e Scripts

BluePlant é um sistema SCADA que suporta as linguagens do Microsoft .NET em completa integração com o Microsoft .NET Framework. Os scripts e lógicas do projeto podem ser escritos em C# ou VB.NET, e um conversor de linguagem integrado permite alternar dinamicamente o código criado entre as linguagens.

No ambiente do BluePlant, é possível compilar, realizar a referência cruzada dos objetos e acessar diretamente (usando o Intellisense) as classes .NET e objetos do projeto, incluindo alarmes, relatórios e nós de comunicação.

As linguagens .NET fornecem um ambiente mais poderoso quando comparado com o VBA ou VBScript, que são linguagens interpretadas e não compiladas. Essas tecnologias deixam muitos erros de programação que são encontrados somente quando projeto em VBA ou VBScript é executado em tempo real, resultando muitas vezes em consequências e resultados indesejados. O ambiente gerenciado do Microsoft .NET Framework dá suporte para localizar e se recuperar das exceções, proporcionando assim um ambiente altamente confiável para o sistema do Runtime e as aplicações.

	Linguagens .NET e Scripts
Criação de funções VB.NET e procedimentos	Acessar objetos no BluePlant diretamente a partir do código
Execução de scripts em eventos e programação	Fácil conexão com tags e eventos do processo usando scripts
Suporte a bibliotecas de classe	Criação de classes próprias acessíveis de outros scripts e telas
Editor .NET integrado com Intellisense	Auxiliar na seleção de nomes da tag e propriedades de objetos
Suporte a exceções e mensagens de rastreamento	Todo o .NET Framework, componentes externos e serviços são facilmente integrados

Tabela 3-2. Linguagens .NET e Scripts

Alarmes e Segurança

Podem ser definidos vários níveis de alarme para cada ponto ou Tag e toda uma gama de comportamentos, tais como registro, reconhecimento, exibição, etc. os quais são pré-definidos para facilitar a configuração. O sistema de segurança pode definir níveis de acesso para cada objeto de tela. Condições de alarme e de segurança são automaticamente replicadas nas aplicações redundantes.

	Alarmes e Segurança
Múltiplas condições de alarme	Hi, HiHi, Lo, LoLo, taxa de alteração e desvio
Alta resolução	Intervalo da estampa de tempo em milissegundo (quando disponível), usando o tempo remoto de E/S, não a hora do computador
Objetos de visualização integrados	Objeto gráfico online e histórico, quando executado localmente ou na Web
Grupo de alarme e item de objetos	Acessar as propriedades dos alarmes diretamente, por exemplo, "total de alarmes ativos", sem a necessidade de criar tags na aplicação

Tabela 3-3. Alarmes e Segurança

Tendência e Historiadores

O BluePlant permite criar arquivos de historiador em bancos de dados externos, tais como Microsoft SQL Server ou Oracle, ou mesmo usar o banco de dados SQL integrado. Ele possibilita salvar os dados com base na alteração de dados ou disparadores de grupo e tem uma opção exclusiva de faixa de tempo que impede o registro de dados com uma estampa de tempo menor do que um valor pré-definido, permitindo a criação de bancos de dados mais compactos. O acesso a um servidor de PI da OSIsoft também é possível.

O recurso de estampa de tempo pode utilizar um valor fornecido diretamente a partir da E/S remota, ao invés do computador, garantindo maior precisão no evento. A organização das amostras permite incluir ou remover tags do registro sem perder a compatibilidade com os últimos dados. Um objeto completo do tipo gráfico de tendência também é fornecido para a visualização de dados online e históricos.

	Tendência e Historiadores
Conexão com banco de dados ADO	Informações do historiador podem ser salvas em qualquer base de dados externa com suporte ADO.NET
Banco de dados SQL integrado	Quando não definidos como um banco de dados externo, o log acontece no banco de dados SQL integrado
Alta resolução	Intervalo da estampa de tempo em milissegundos (quando disponível), usando o tempo de E/S remoto, não a hora do computador
Disparador por Tag ou grupo	Permitir o salvamento de um registro de acordo com a alteração do Tag ou baseado em eventos de processo
Banda morta do historiador por Tag	Permitir a definição de uma variação mínima de Tag para disparo do registro
Configuração do intervalo de tempo mínimo	Permitir a definição de um intervalo mínimo para gravação, permitindo a criação de bancos de dados mais compactos
Tabelas de banco de dados com múltiplos tags	Permitir a criação de um grupo de tags e armazena-os na mesma tabela de dados para acelerar a gravação e carregamento
Objeto de visualização de tendência integrado	Objeto gráfico de histórico e online executado localmente ou na Web

Tabela 3-4. Tendência e Historiadores

Dispositivos e Redes

O BluePlant é fornecido com um driver OPC DA para coleta de informações de dispositivos remotos. Além do OPC, o BluePlant também oferece suporte a drivers de comunicação personalizados para acessar diretamente CPs, sistemas remotos de E/S, barramentos de campo padrões, laços únicos e múltiplos, scanners, leitores de código de barras, dispositivos RFID e monitores digitais.

A ferramenta de configuração de dispositivo pode importar bancos de dados de servidores OPC, arquivos de texto ou CSV. Se o dispositivo for compatível, automaticamente implementa sistemas multitarefa em redes TCP/IP ou cenários multi-seriais. A sintaxe de endereçamento segue a convenção de nomenclatura do dispositivo remoto, tornando a configuração e a manutenção muito mais fáceis. Um conjunto completo de ferramentas de desempenho e diagnóstico está incluído também.

	Dispositivos e Redes
Importação de configurações de pontos de dados	Copiar e colar do Microsoft Excel, importar bancos de dados de servidores CSV ou OPC
Comunicação executada em processo isolado	Proteção total para o ambiente do Runtime e desempenho avançado em processadores multi-core
Fácil comunicação com vários canais	Criar automaticamente múltiplas tarefas nos protocolos multi-serial ou TCP/IP
Nomeação abstrata para nós e estações	Fornecer maneira fácil de renomear e manter o endereço IP e a configuração de rede de E/S
Criação dinâmica de blocos otimizados	Seleção simples de pontos de leitura e escrita, baseada no protocolo, e criação de blocos otimizados
Configuração de pontos segue a sintaxe do dispositivo	Endereçar pontos do dispositivo, esse endereçamento é usado nas ferramentas de programação do CP
Propriedades de canais e nós	Acessar as propriedades diretamente, por exemplo, status do nó e tags da aplicação não são necessários
Customização de eventos de escrita	Configuração fácil para comandos e eventos, escrita de todos os eventos ou apenas no caso de alteração de valor

Tabela 3-5. Dispositivos e Redes

Conjunto de Dados

O módulo de conjunto de dados incluído no BluePlant fornece uma interface de fácil operação para a troca de dados em tempo real com bancos de dados externos, XML, CSV ou arquivos de texto, bem como a possibilidade de acessar tabelas e consultas SQL.

Para os mais populares bancos de dados e fontes de dados (Microsoft SQL Server, Oracle, Arquivos CSV, Microsoft Access, PI, Firebird, Informix e Excel), o BluePlant fornece configurações pré-definidas que reduzem o gerenciamento de configurações a um clique do mouse. Qualquer banco de dados que ofereça suporte a ODBC, ADO.NET ou OLE-DB também pode ser acessado. Um banco de dados DB SQL integrado é fornecido como opção de banco de dados local para a aplicação.

Os dados coletados via conjunto de dados podem ser mapeados dinamicamente em tempo real para pontos/tags e podem ser usados em scripts ou relatórios ou ainda apresentados em telas usando um poderoso objeto visual de grade de dados.

	Conjunto de dados
Acessar textos, CSV e arquivos XML	Definir a ligação em tempo real com tags e conteúdo de arquivo
Definição de múltiplas fontes de banco de dados	Gerenciar facilmente múltiplas conexões de banco de dados
Mapeamento de tags com tabelas de dados	Utilitário de configuração de alto nível para gerenciar as tabelas de banco de dados usadas no projeto
Definição de consultas e mapeamentos	Gerenciar várias consultas disparadas por eventos de processo e condições de filtro usando pontos de dados em tempo real
Poderoso objeto de visualização de grade de dados	Abrangente e poderoso objeto de grade de dados para criar interfaces de usuário tanto locais quanto na Web
Propriedades de tabela e consultas	Acessar propriedades (por exemplo, contagem de linhas) diretamente, sem a necessidade de criação de tags na aplicação

Tabela 3-6. Conjunto de Dados

Relatórios

BluePlant oferece suporte a serviços Web, XML e outras interfaces de troca de dados com o objetivo de fornecer dados para as ferramentas de relatórios externas. Em contraste com outros pacotes, onde os relatórios são necessariamente criados em outra ferramenta, o BluePlant tem seu próprio editor de relatório interno.

O editor de relatório permite a inclusão de texto dinâmico, símbolo gráfico dinâmico e gráficos, e pesquisas em banco de dados, em um editor completo e de fácil utilização. Os relatórios podem ser salvos nos formatos texto, HTML, PDF ou XPS e facilmente apresentados em clientes remotos e telas Web.

	Relatórios
Editor integrado	Editor de texto fácil de usar, permitindo a inclusão de tabelas, imagens, hiperlinks e formatação de texto
Suporte a texto, HTML e XPS	Salvar relatórios em vários formatos, como XPS, por exemplo, permitindo fácil implantação em ambientes distribuídos
Copiar e colar	Editar no Microsoft Word ou em editor HTML ou RTF, basta copiar e colar o conteúdo do BluePlant
Tags em tempo real facilmente inseridas	Com apenas um clique, o usuário pode adicionar valores de dados em tempo real aos relatórios

Tabela 3-7. Relatórios

Nota:

Não é possível fazer relatórios baseados em resultados dinâmicos a partir do banco de dados, ou seja, a pesquisa deve sempre retornar um número exato de linhas.

Telas do Cliente

O editor gráfico integrado no BluePlant utiliza a tecnologia WPF da Microsoft para permitir a criação de interfaces completas de usuário com mapeamento em tempo real de valores e tags do processo: um poderoso e completo conjunto de animações dinâmicas também está incluído.

As telas são salvas internamente usando XAML, o qual fornece independência de resolução, isolamento do código e fácil extensibilidade. Uma biblioteca de símbolos, onde os símbolos também podem manter um vínculo dinâmico com a biblioteca, acelera criação de processos sinóticos. Todas as tecnologias do cliente oferecem suporte de servidor redundante.

Três tecnologias são utilizadas em clientes remotos:

- BluePlant Visualizer Clients: é executado como uma aplicação desktop e permite bloquear o chaveamento de tarefas do Windows (teclas CTRL+ALT+DEL e ALT+TAB do Windows). Isto é ideal para operadores/usuários de intranet com demandas de segurança
- Web Smart Clients: utiliza a tecnologia .NET Smart Client da Microsoft e em clientes remotos é possível instalar com um único clique e sem requisitos de administrador. A aplicação é atualizada automaticamente nos clientes remotos quando atualizado no servidor. O BluePlant usa todo o potencial do computador remoto e ainda mantém as vantagens de uma instalação centralizada
- Web XBAP Partial Trust Clients: as telas do cliente podem ser executadas diretamente a partir de navegadores da Web, sem necessidade de instalação de qualquer software (inclusive controles do ActiveX). O Partial Trust Clients garante que as telas do cliente sejam executadas em um ambiente completamente isolado. Assim como nos Smart Clients, quando o aplicativo é atualizado no servidor, também é atualizado automaticamente nos clientes

Objetos do Runtime

Mais avançada do que a maioria dos sistemas, onde é necessário criar tags ou variáveis para todas as propriedades internas e customizar lógicas para projetos, o BluePlant permite que as aplicações acessem diretamente todos os objetos do Runtime criados no projeto.

Isso significa que tags temporários não são necessários para gerenciar o status dos nós da rede do CP, do número total de alarmes em um grupo ou do número de linhas em um conjunto de dados. É possível acessar os objetos do Runtime (que representem um nó da rede), um grupo de alarme ou conjunto de dados e exibir as informações necessárias ou tomar uma ação diretamente através das propriedades integradas.

Isolação de Módulo

Para um melhor desempenho, segurança e confiabilidade, os módulos da UCP com maior consumo e sensibilidade, tais como scripts, conjuntos de dados, dispositivos (drivers de comunicação), relatórios e telas, são executados em seus próprios processos ou no domínio da aplicação em sua própria tarefa, independentemente do banco de dados em tempo real do servidor.

Além das vantagens descritas anteriormente, a arquitetura do BluePlant também permite a distribuição da aquisição de dados da aplicação, ou qualquer aplicação intensiva da UCP em diferentes computadores em um ambiente distribuído, proporcionando uma maior flexibilidade para implementar vários cenários redundantes e consequente simplificação da manutenção em campo.

Ferramentas do Runtime e Diagnóstico

A ferramenta de monitoração de propriedade permite verificar e simular valores em todos os módulos e objetos e também iniciar e parar todos os módulos individualmente.

A ferramenta de janela de rastreamento automaticamente gera mensagens de sistema sobre eventos importantes no Runtime e pode ser facilmente estendida para emitir mensagens específicas relacionadas com eventos de script, alterações de dados de tags/pontos ou ações do usuário.

A ferramenta de informações do módulo é uma ferramenta de desempenho e perfil avançado que fornece informações internas de todo o ambiente do Runtime.

	Ferramentas de Diagnóstico e do Runtime
Modo teste	Executar projetos com proteção, como, somente leitura em dispositivos externos ou arquivos temporários no historiador
Informações do módulo	Ferramentas avançadas para perfil de desempenho e diagnósticos de sistemas internos
Ferramenta de localização	Criar interface de usuário do operador em várias linguagens e, dinamicamente, alterna entre elas no Runtime
Janela de rastreamento	Ao criar uma aplicação esta ferramenta fornece monitoração de Tag e mensagens de diagnóstico do sistema
Monitoração de propriedade	Verificar e simular valores de Tag e propriedades, iniciar e parar módulos funcionais

Tabela 3-8. Ferramentas do Runtime e de Diagnóstico

Implantação e Teste de Projeto

Antes de executar uma aplicação ou projeto, pode ser utilizado o exclusivo “Modo Teste” do BluePlant o qual pode executar o projeto ou a aplicação em um ambiente de teste seguro. No “Modo Teste”, não é possível enviar comandos para os controladores remotos (apenas comandos de leitura são enviados), alarmes e historiadores salvam os dados em arquivos temporários e os bancos de dados externos de tempo real são acessados em modo somente leitura.

Após concluir com sucesso o teste, é preciso executar a opção "Inicializar" para obter a funcionalidade completa. Quando o projeto ou aplicação estão prontos para serem implantados no campo, deve ser utilizado o recurso “Publicar” para configurar as opções de redundância (se aplicável) e para criar uma cópia do projeto somente leitura, versão controlada, cópia do projeto para a instalação em campo.

	Ferramentas de Implantação e Teste de Projeto
Abertura de vários projetos	Abrir vários projetos simultaneamente no computador
Engenharia remota	Acessar e editar remotamente a configuração do projeto
Execução como serviço do Windows	Executar o Runtime no servidor, instalado como um serviço do Windows
Proteção de chaveamento de aplicações	Proteger contra chaveamentos de aplicações não autorizadas em interfaces do operador usando as teclas CTRL+ALT+DEL do Windows ou outras
Atalhos de inicialização	Utilizar atalhos simples de inicialização e parâmetros para customização de inicialização
Projeto de arquivo único e recursos embarcados	A configuração de todo o projeto é salva em um único arquivo protegido, incluindo todas as imagens e bitmaps usados em telas e relatórios

Tabela 3-9. Ferramentas de Implantação e Teste de Projeto

Drivers de Comunicação

Estão disponíveis vários drivers para os principais fabricantes de CPs e sistemas de automação.

	Drivers de Comunicação
Altus	ALNET I
	ALNET II
	FBS
MODBUS	RTU-TCP
	RTU-TCP Escravo
Allen Bradley	CIP (Control Logix)
OPC	DA
	XML-DA
	UA
	Xi
Siemens	S7 Ethernet

Tabela 3-10. Drivers de Comunicação

4. BluePlant

Requisitos Mínimos para Instalação e Operação

Os modelos BluePlant Student, BluePlant Express e BluePlant Lite apresentam quatro canais disponíveis e o modelo BluePlant Enterprise conta com 64. As tabelas a seguir mostram os requisitos mínimos para instalação e operação do BluePlant usando diferentes quantidades de canais.

	BluePlant Express, BluePlant Student, BluePlant Lite e BluePlant Enterprise (até 4 canais)
Plataforma	PC com Windows XP® (32 bits), Windows Vista® (32 bits), Windows 7® (32 bits ou 64 bits), Windows 8® (32 bits ou 64 bits), Windows Server2008 ou Windows Server2012
Processador	Intel Core 2 Duo (mínimo)
Espaço em disco	1 Gbyte (mínimo), 2 Gbytes (recomendado)
Memória RAM	2 Gbytes (mínimo), 4 Gbytes (recomendado)
Resolução	1024 x 768 (mínimo), 1280 x 1024 (recomendado)
Idioma	Qualquer idioma

Tabela 4-1. Configuração para até 4 Canais

	BluePlant Enterprise (até 8 canais)
Plataforma	PC com Windows XP® (32 bits), Windows Vista® (32 bits), Windows 7® (32 bits ou 64 bits), Windows 8® (32 bits ou 64 bits), Windows Server2008 ou Windows Server2012
Processador	Intel Core i5 (mínimo)
Espaço em disco	1 Gbyte (mínimo), 2 Gbytes (recomendado)
Memória RAM	4 Gbytes (mínimo), 6 Gbytes (recomendado)
Resolução	1024 x 768 (mínimo), 1280 x 1024 (recomendado)
Idioma	Qualquer idioma

Tabela 4-2. Configuração para até 8 Canais

	BluePlant Enterprise (até 16 canais)
Plataforma	PC com Windows XP® (32 bits), Windows Vista® (32 bits), Windows 7® (32 bits ou 64 bits), Windows 8® (32 bits ou 64 bits), Windows Server2008 ou Windows Server2012
Processador	Intel Core i7 (mínimo)
Espaço em disco	1 Gbyte (mínimo), 2 Gbytes (recomendado)
Memória RAM	6 Gbytes (mínimo), 8 Gbytes (recomendado)
Resolução	1024 x 768 (mínimo), 1280 x 1024 (recomendado)
Idioma	Qualquer idioma

Tabela 4-3. Configuração para até 16 Canais

Nota:

Plataforma: É necessária a instalação do Microsoft .NET Framework 4.0.

ATENÇÃO:
A quantidade de canais e o desempenho da aquisição de dados são os principais motivos para selecionar os requisitos de operação. Se forem necessários mais de 16 canais ou se existirem outras demandas específicas do projeto, recomenda-se fortemente contatar o Suporte Técnico da Altus através do site www.altus.com.br ou e-mail altus@altus.com.br.

Instalação

Para executar a instalação do software BluePlant é necessário realizar o download do arquivo de instalação a partir do site www.altus.com.br. Em seguida, devem ser fechados todos os programas em execução no PC, clicar duas vezes no arquivo de instalação e depois em Next.

A tela do contrato de licença que aparecerá deve ser lida atentamente. Caso os termos da licença sejam aceitos, selecionar a opção para continuar a instalação. A tela de instalação a seguir irá aparecer, para prosseguir clicar no botão Next.

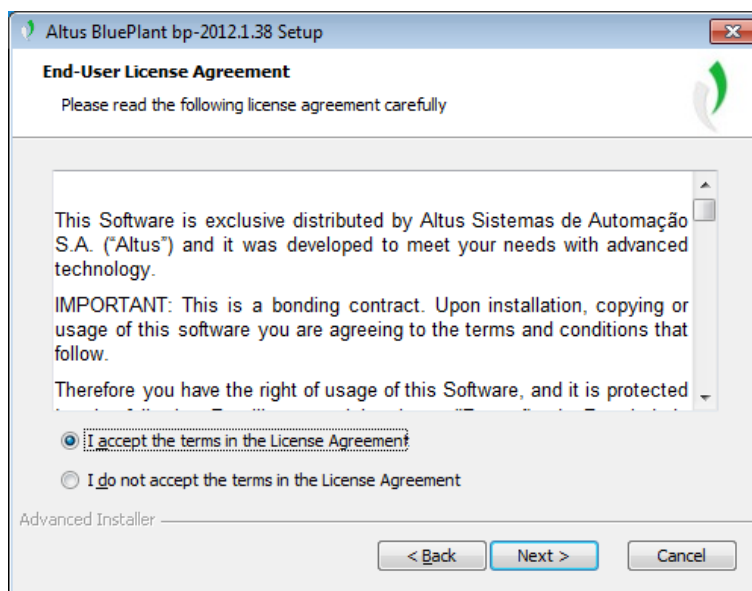



Figura 4-1. Tela do Contrato de Licença do BluePlant

Durante a instalação, aparecerá outra tela de licença, a qual deve ser lida atentamente. Este acordo é do software Advosol OPC Core Components. Este software oferece um suporte completo na plataforma .NET para todos os principais padrões OPC. OPC é um conjunto de especificações normatizadas. Tais padrões resultaram da colaboração de vários dos principais fornecedores de automação em todo o mundo trabalhando em cooperação com a Microsoft. A especificação definiu um conjunto padrão de objetos, interfaces e métodos para uso em controle de processos e aplicações de automação para facilitar a interoperabilidade. Atualmente, existem centenas de clientes e servidores OPC Data Access.

Caso os termos sejam aceitos, a caixa de verificação deve ser marcada para continuar a instalação. A próxima tela da instalação irá aparecer, para prosseguir, clicar no botão Next.

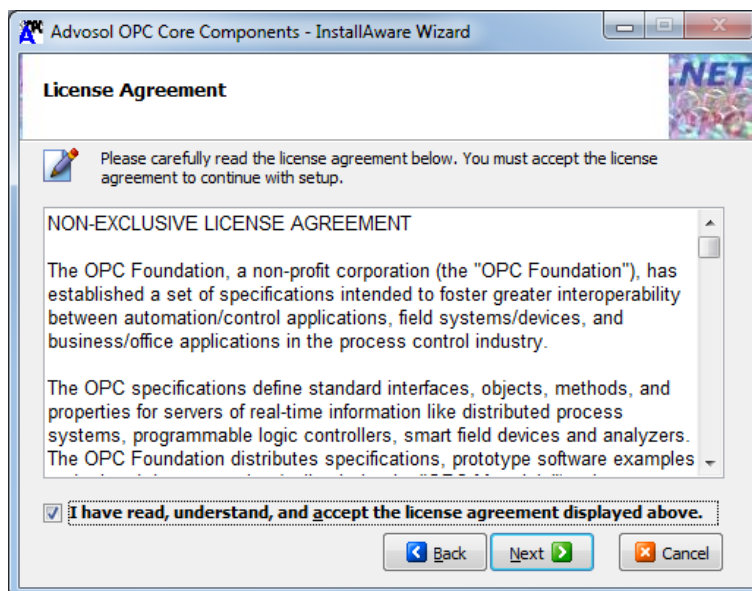



Figura 4-2. Tela do Contrato de Licença do Advosol

A tela de instalação do Advosol OPC Core Components deve aparecer. Aguardar enquanto os arquivos necessários são instalados no computador. Isso pode levar alguns minutos dependendo da configuração do computador.

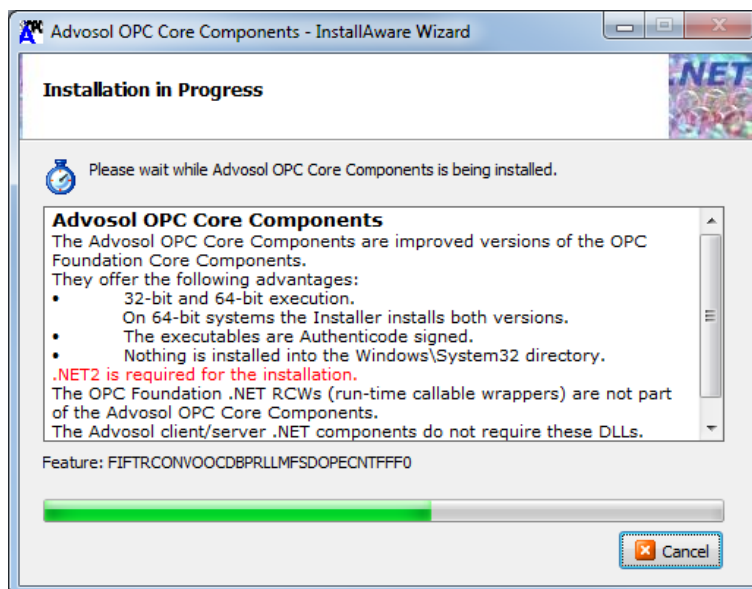



Figura 4-3. Tela de Instalação do Advosol

Após a conclusão da instalação do Advosol OPC Core Components, inicia a instalação do BluePlant. Aguardar enquanto os arquivos necessários são instalados no computador. Isso pode levar alguns minutos dependendo da configuração do computador.

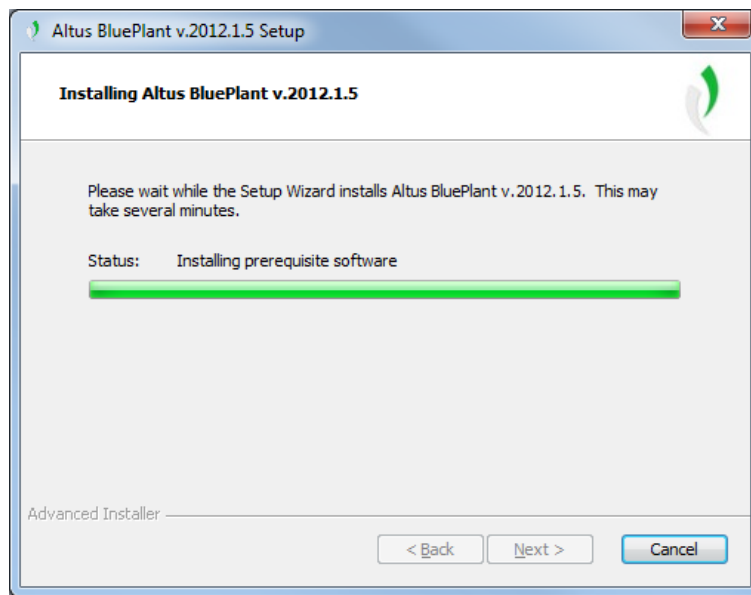



Figura 4-4. Tela de Instalação do BluePlant

Após a conclusão da instalação, a tela a seguir será exibida. Clicar no botão Finish para concluir o procedimento de instalação.

Figura 4-5. Tela de Encerramento de Instalação do BluePlant

O BluePlant está instalado e pronto para usar. Para executá-lo, clicar no atalho "BluePlant", criado durante a instalação, no Menu Iniciar.

Licenças e Hardkey

O BluePlant possui alguns modelos onde suas características são habilitadas através de licenças presentes em uma hardkey. A hardkey é um meio físico onde estão presentes as licenças requisitadas e para que o BluePlant libere as funcionalidades presentes na hardkey, a mesma deve ser conectada ao servidor onde será executado o projeto.

Na Tabela 2-1, podem ser conferidos os modelos BluePlant disponíveis e exceto o modelo Express, todos demais são habilitados por licenças presentes em hardkey.

Uma vez que a hardkey é conectada no servidor, as informações podem ser visualizadas através da guia "License" conforme Figura 4-9.

> ATENÇÂO:
> Devem ser tomados cuidados para que a hardkey não seja danificada, pois nestes casos se o acesso às informações presentes na hardkey for comprometido, as funcionalidades habilitadas não mais estarão disponíveis. Neste caso deve-se entrar em contato com o Suporte Técnico da Altus através do site www.altus.com.br ou e-mail altus@altus.com.br.

Iniciando com o BluePlant

Uma vez instalado o BluePlant no computador, um duplo clique no ícone apresentado abaixo, permite abrir a ferramenta.

Figura 4-6. Ícone do BluePlant

Gerenciamento de Projeto

Após a inicialização do BluePlant deve aparecer a janela de Gerenciamento do Projeto.

No centro da tela inicial, em sua parte superior, encontram-se as guias denominadas "Projects", "Server", e "License"

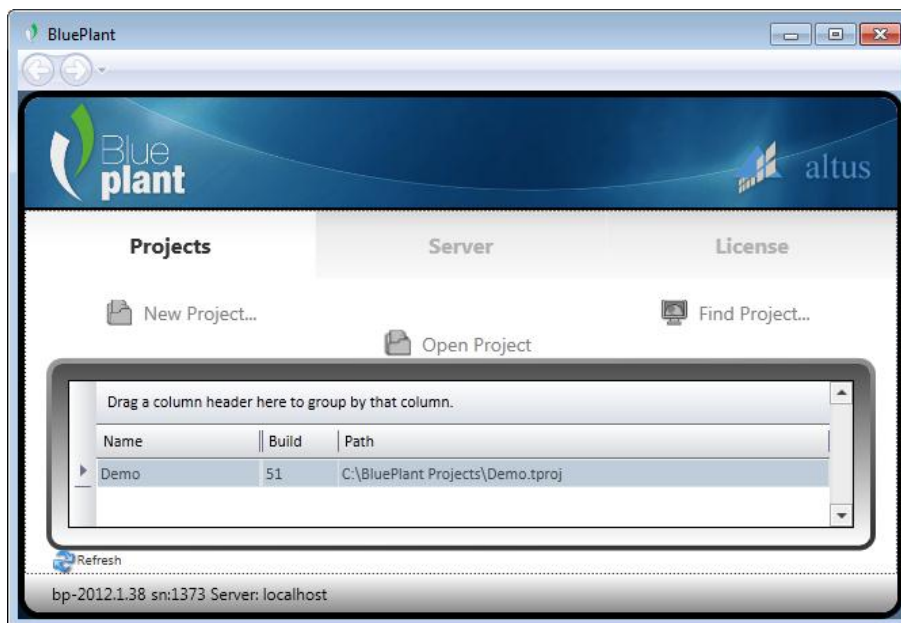



Figura 4-7. Tela de Gerenciamento de Projeto

Em seguida aparece a estrutura de menus padrão Web do BluePlant. Este design permite um acesso mais eficiente ao leque de ferramentas, reduzindo a quantidade de cliques. Além disso, uma tabela lista os projetos que podem ser acessados.

Na guia “Projects”, estão os ícones para criar um projeto “New Project”, para abrir um projeto selecionado “Open Project...” e para localizar outros projetos “Find Project...” que não estão localizados no diretório padrão do BluePlant.

A guia “Server” tem a aparência mostrada na Figura 4-8 onde duas opções estão disponíveis: Localhost e Remote. A primeira significa que o projeto será executado localmente e a segunda que o projeto será executado a partir de um servidor remoto.

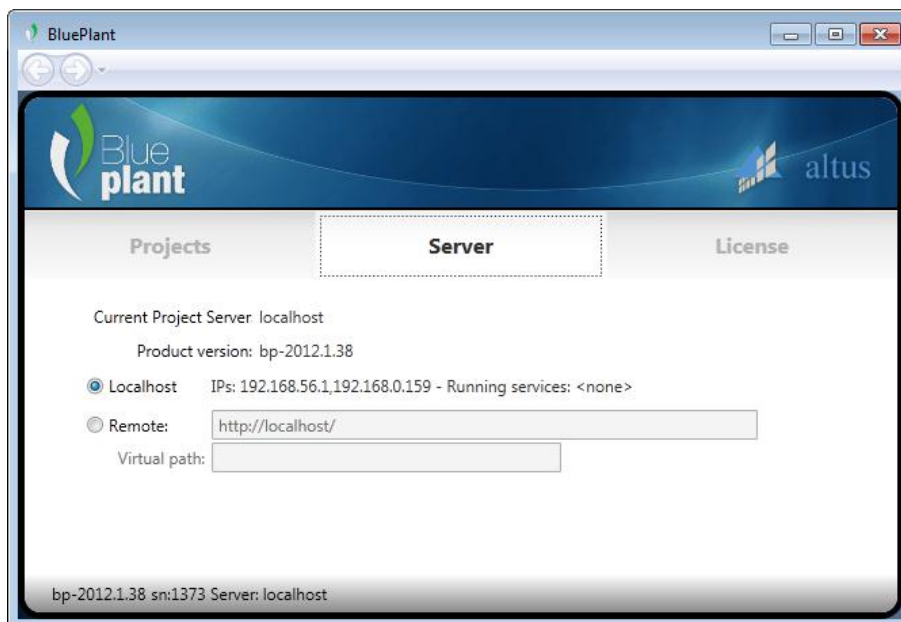



Figura 4-8. Guia Servidor

A terceira guia refere-se à licença do produto, incluindo o modelo, a versão do produto, família, número de série, código do produto entre outras informações presentes na hardkey como pode ser visto na Figura 4-9.

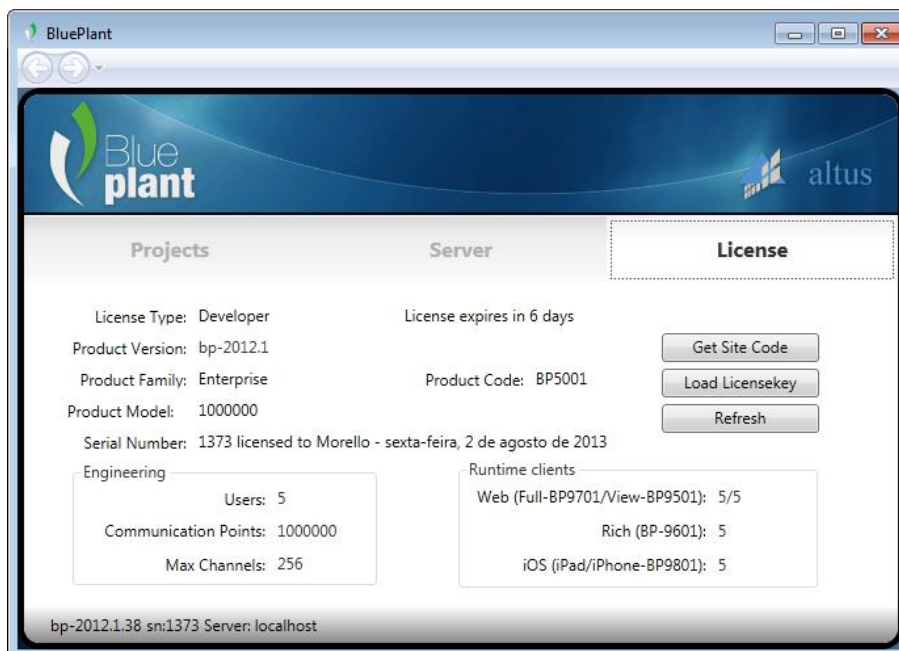



Figura 4-9. Guia Licenças

Primeiros Passos com o BluePlant

Esta seção apresenta uma sequência de operações para a criação de um projeto simples ou abertura de um projeto existente. Junto com a ferramenta é disponibilizado um demo que permite uma visão simplificada dos recursos disponíveis no BluePlant. O detalhamento das funcionalidades ilustradas aqui será explorado no decorrer deste manual.

Selecionando o Projeto Demo

A partir da guia “Projects...” no editor de projetos, pode ser selecionado o projeto Demo.tproj, o qual contempla uma visão geral das funcionalidades e recursos do BluePlant.

Para abrir o projeto, basta selecionar o projeto presente na lista e clicar em “Open Project”.

Criando um Novo Projeto

As configurações básicas do projeto a ser criado são definidas a partir da janela de abertura do BluePlant clicando em “New Project...” que pode ser visto na Figura 4-7.

Na janela seguinte, apresentada na Figura 4-10, estão indicados os passos para criar um novo projeto.

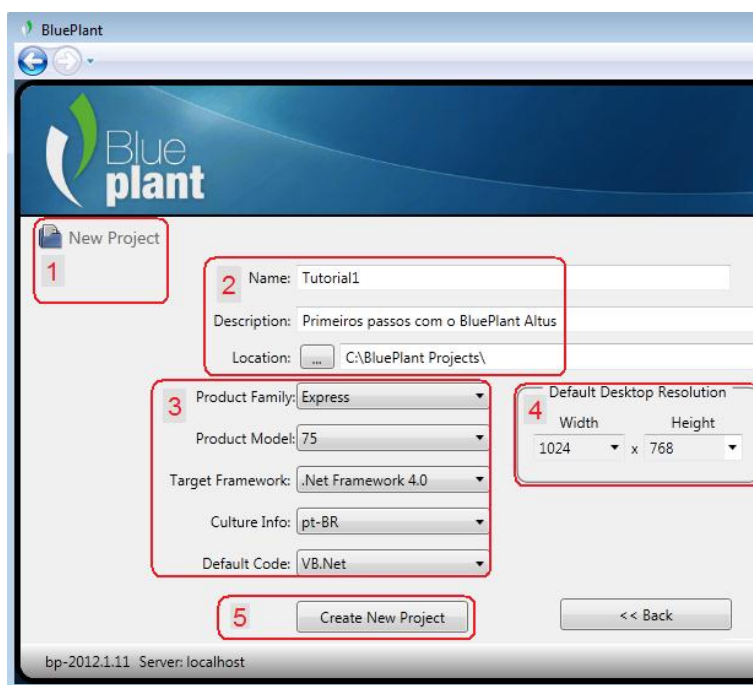



Figura 4-10. Criando um Novo Projeto

1. Indicação de novo projeto
2. Definição do nome do projeto, descrição e diretório onde será gerado o projeto
3. Configurações iniciais tais como: família e modelo do produto, plataforma, cultura e código (script) padrão
4. Ajustes da resolução da tela
5. Botão para criar um novo projeto e iniciar o desenvolvimento da aplicação

O novo projeto criado está referenciado na lista da janela de abertura do BluePlant.

Interface do Usuário

A configuração da interface do usuário opera como uma página da Web front-end onde se pode facilmente navegar entre os itens e editá-los. Os dados são salvos automaticamente no back-end, sem a necessidade de abrir/salvar/fechar os menus. A interface do usuário contempla, à esquerda, o menu principal no topo e o submenu logo abaixo o qual se altera quando se seleciona outro item do menu principal. Além disso, à direita, estão as guias da área de trabalho e, mais acima, uma barra de links rápidos e itens recentes. A Figura 4-11 ilustra a interface do usuário.

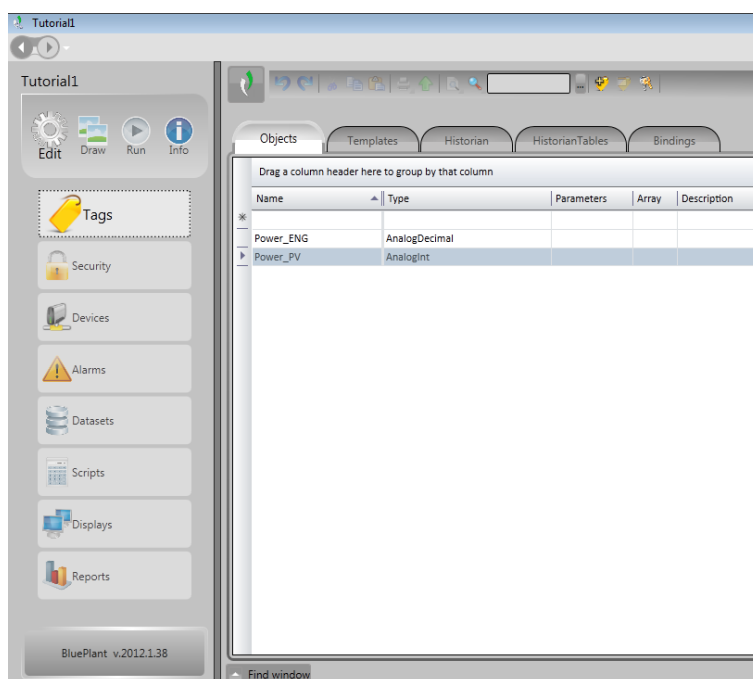



Figura 4-11. Interface do Usuário

Criando Tags no Projeto

No ambiente de edição é possível editar objetos como, por exemplo, tags. A Figura 4-12 mostra o caminho para a edição de tags. Neste exemplo, foram criados dois tags analógicos: o valor bruto de um medidor de potência elétrica a ser lido do controlador programável denominado de Power_PV e o valor de engenharia desta medição chamado de Power_ENG.

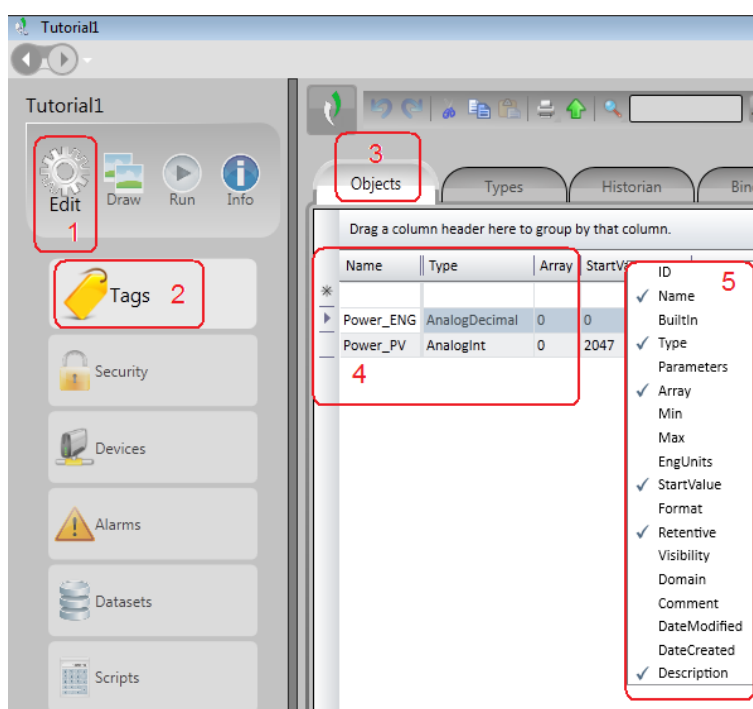



Figura 4-12. Editando Tags do Projeto

Os passos para a criação desses tags estão listados a seguir conforme numeração indicada na figura anterior.

1. Acessar o menu Edit
2. Selecionar a opção Tags
3. Acessar a aba Objects na qual aparecerá a tabela com os tags do usuário
4. Digitar os dados relativos ao novo tag da aplicação na linha marcada com um asterisco (*). Para confirmar a inclusão na tabela deve ser pressionada a tecla ENTER
5. Selecionar as opções de configuração dos tags através das colunas da tabela desejadas clicando com o botão direito do mouse no cabeçalho de uma coluna e marcando os itens para exibição

Também é possível criar um novo tag acessando a barra de ferramentas no topo do ambiente de edição. Basta clicar no ícone e configurar os parâmetros do tag.

A Figura 4-13 mostra o processo de criação de um tag vetor de uma dimensão, com três posições do tipo analógico inteiro a partir do Menu Editar. Um tag vetor é um conjunto de tags com o mesmo nome, cuja identificação se dá através de índices. Neste exemplo, o acesso em outros locais da aplicação a esse tag, se dá via sintaxe TT[0], TT[1], TT[2] e TT[3].

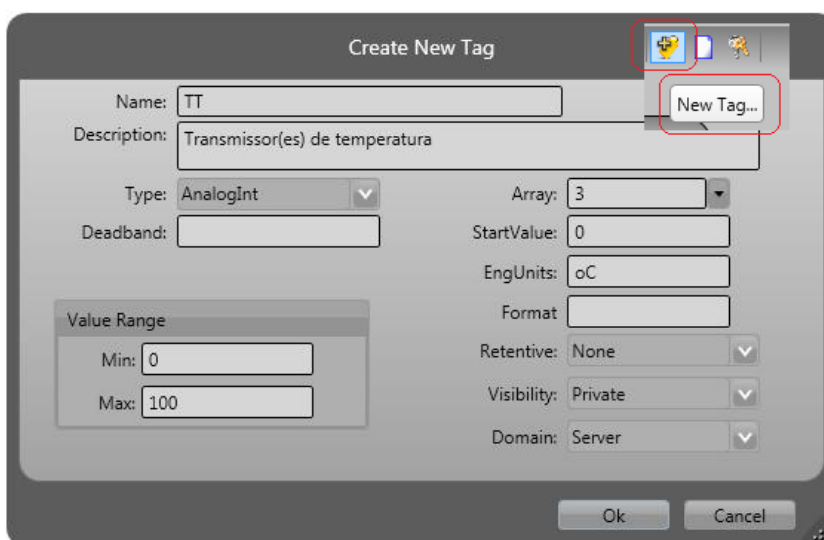



Figura 4-13. Processo de Criação de Tags

Associando tags a um protocolo comunicação

Para realizar a associação das tags criadas a um protocolo de comunicação, os passos descritos a seguir devem ser executados.

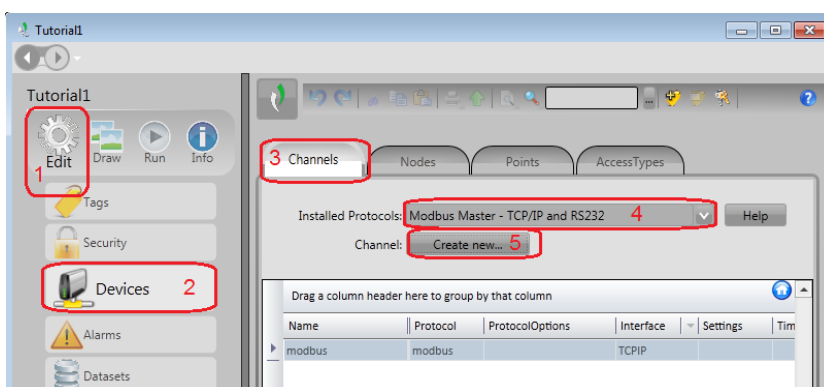



Figura 4-14. Processo de Criação de Canal de Comunicação

1. Acessar o menu Edit
2. Selecionar a opção Devices
3. Acessar a guia Channels na qual deve aparecer o campo com os protocolos disponíveis
4. Selecionar no campo Installed Protocols o protocolo de comunicação a ser utilizado
5. Para confirmar a criação do canal, clicar no botão “Create new...”

Uma vez o protocolo selecionado e confirmada a criação do canal, uma janela com as propriedades do protocolo deve aparecer, como pode ser visto na Figura 4-15.

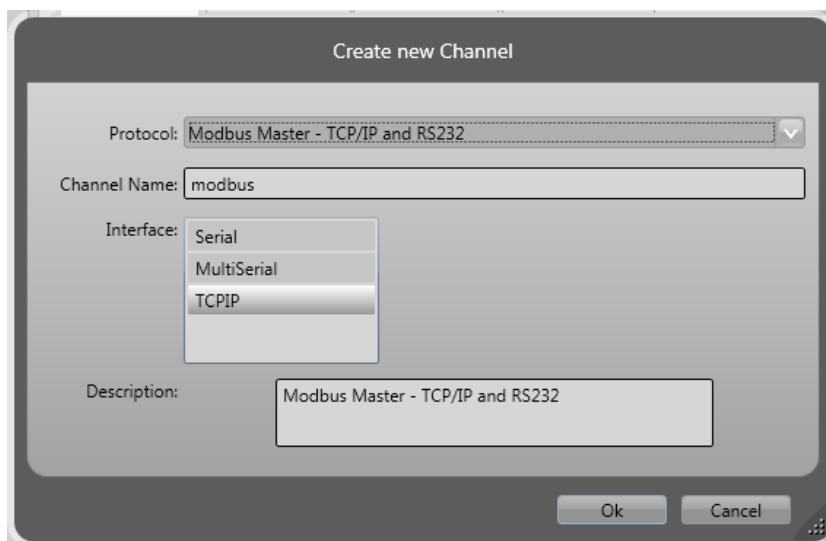



Figura 4-15. Opções de Protocolos para Criação de um Canal

Na guia Nodes é o local onde são criados os nós de acordo com os canais existentes. Para mais detalhes de como criar e configurar nós, pode ser consultado no capítulo Editando Dispositivos.

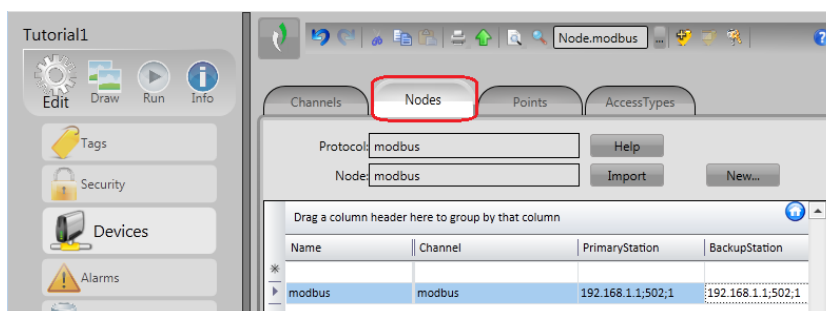



Figura 4-16. Inclusão e Configuração de Nós

Na guia Points é onde são criadas as associações entre os tags, o nó e os endereços do protocolo, além de configurações de tipos de acesso, escalas e tipos de dados.

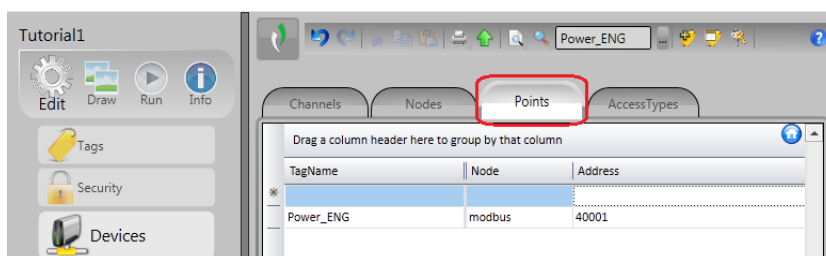



Figura 4-17. Associação de Tags com Nós e Endereços de Protocolos

Ao selecionar o campo TagName da guia Point, deve abrir uma janela como apresentado na Figura 4-18, onde estão todos os tags criados e podem ser selecionados para associar ao nó criado.

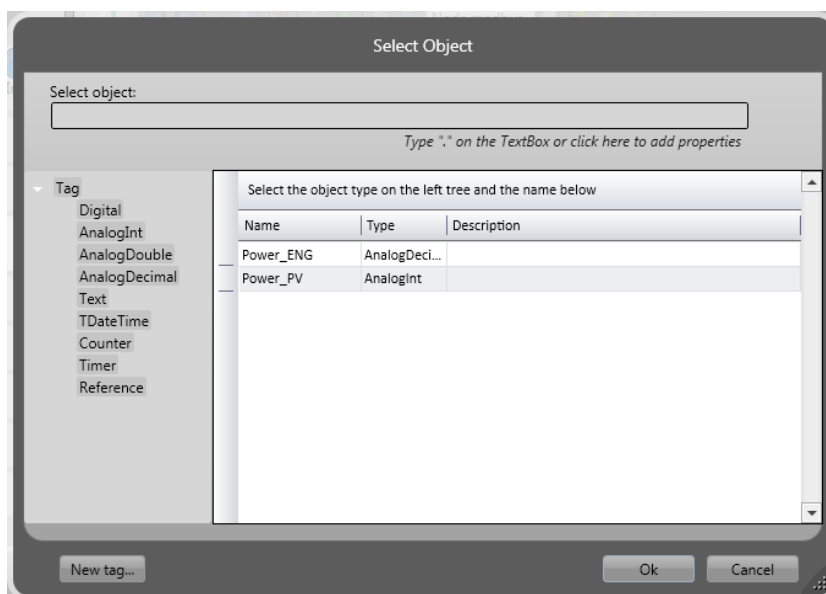



Figura 4-18. Seleção do Tag para Associar a um Nó

Incluindo objetos na tela principal

No menu Draw é possível criar as telas que compõem a aplicação. As telas são formadas por objetos, tais como símbolos e textos.

Vamos explorar a inclusão desses dois tipos de objetos na tela principal padrão.

Inclusão de Símbolo

Para realizar a inclusão de um símbolo do tipo tacômetro na tela principal (MainPage), as seguintes etapas podem ser contempladas:

1. Selecionar Menu Draw
2. Acessar a aba Drawing
3. Abrir a biblioteca de símbolos
4. Selecionar o símbolo do tacômetro, posicionando-o no local desejado
5. Configurar as propriedades do símbolo

Esses passos estão referenciados na Figura 4-19 conforme a numeração correspondente.

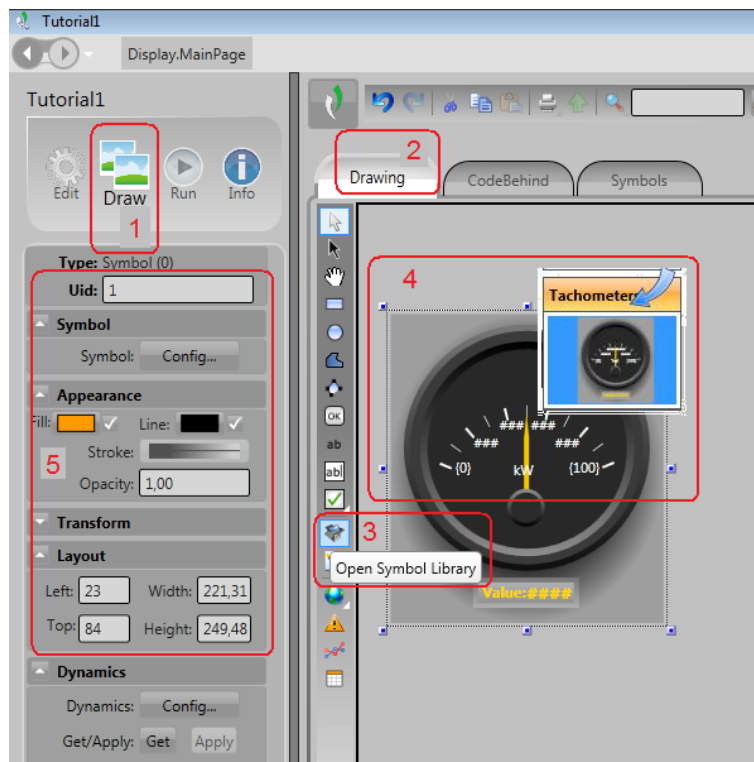



Figura 4-19. Inclusão de um Símbolo do Tipo Tacômetro

Para associar um tag ao símbolo, as seguintes etapas podem ser consideradas:

1. Clicar com o botão direito do mouse no símbolo
2. Selecionar a opção Symbol links

A Figura 4-20 mostra essas etapas de associação de tag.

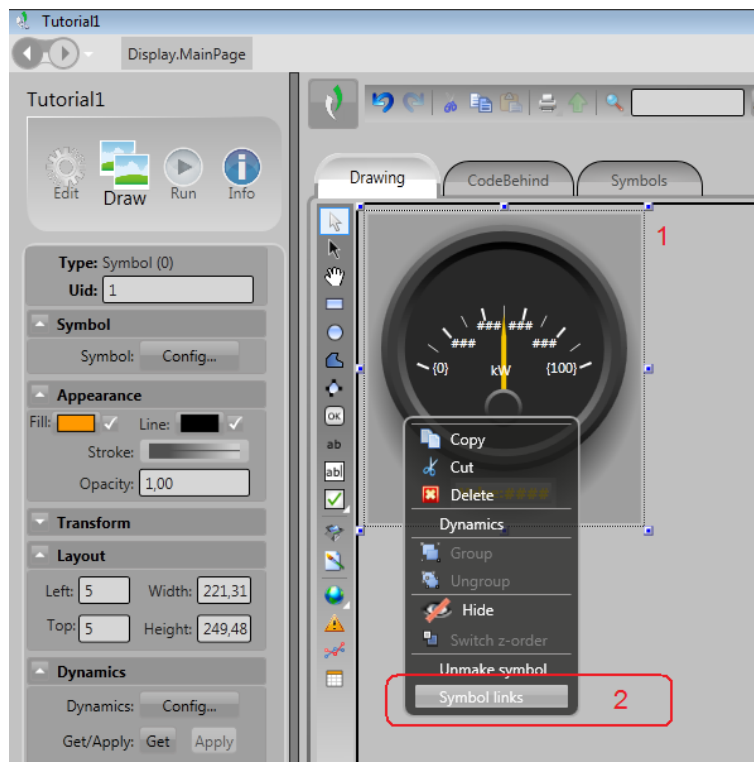



Figura 4-20. Associação de Tag a um Símbolo

Na janela seguinte é possível configurar o tag e seus parâmetros conforme descrito nos passos a seguir:

1. Configurar as propriedades do símbolo (rótulo e valores máximo/mínimo)
2. Associar um tag ao símbolo no campo RotateValue

Neste caso foi utilizado o tag cliente “SimulationAnalog” para simular o movimento do ponteiro do medidor.

A Figura 4-21 mostra essa configuração.

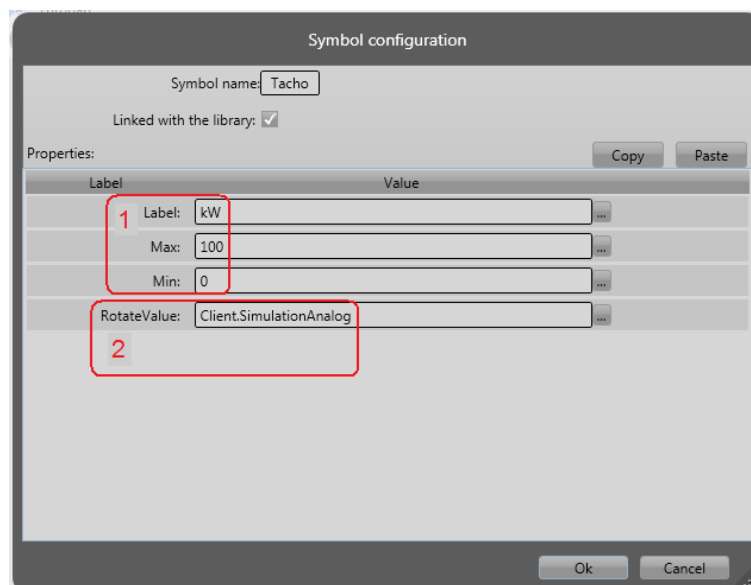



Figura 4-21. Configuração do Tag Associado ao Símbolo

Inclusão de Texto

No menu Draw para incluir um texto na tela podemos proceder conforme segue:

Selecionar a opção de Saída de Texto:

1. Posicionar o texto no local desejado
2. Configurar as propriedades do texto

A Figura 4-22 ilustra esta sequência.

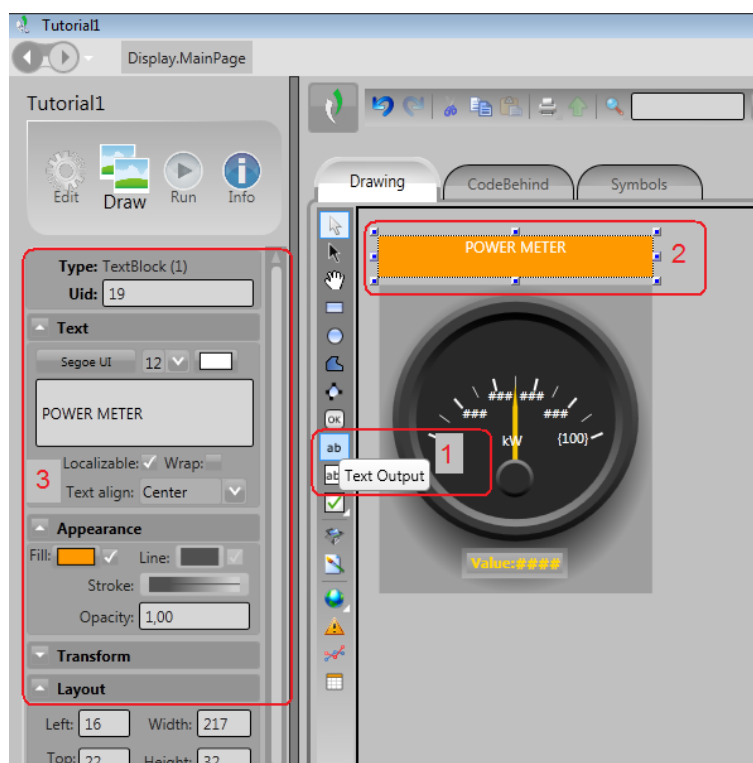



Figura 4-22. Inclusão de um Texto

Executando um Projeto

O ambiente de execução (Run) fornece acesso a todos os recursos de execução do projeto.

Os seguintes módulos estão disponíveis e são explorados nesta seção: Build (Compilar), Test (Testar), Startup (Inicialização) e Publish (Publicar).

Além disso, nesse menu estão incluídas as seguintes ferramentas: UseCount (Contagem de uso), Localization (Localização) e Extensions (Extensões), as quais serão exploradas no decorrer desse manual.

Compilação do Projeto

A compilação verifica o projeto em relação a eventuais erros e otimiza o sistema para operação rápida e eficiente. O tempo necessário para compilar um projeto depende de seu tamanho e a capacidade de processamento do computador. Para compilar o projeto os seguintes passos servem como orientação:

1. Selecionar o menu Run
2. Selecionar a opção Build
3. Acessar a aba Messages
4. Compilar o projeto

5. Confirmar a compilação

A Figura 4-23 ilustra o processo de compilação do projeto.

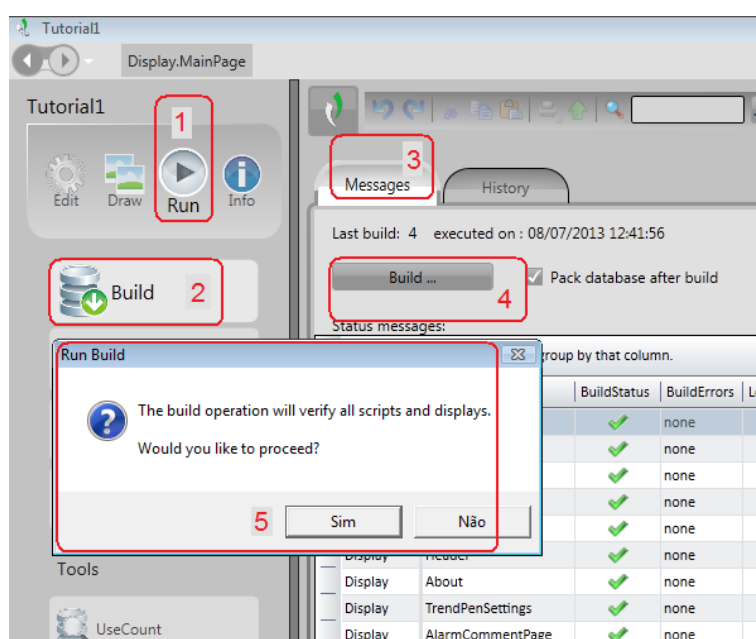



Figura 4-23. Compilação do Projeto

A Figura 4-24 mostra a janela de progresso da compilação.

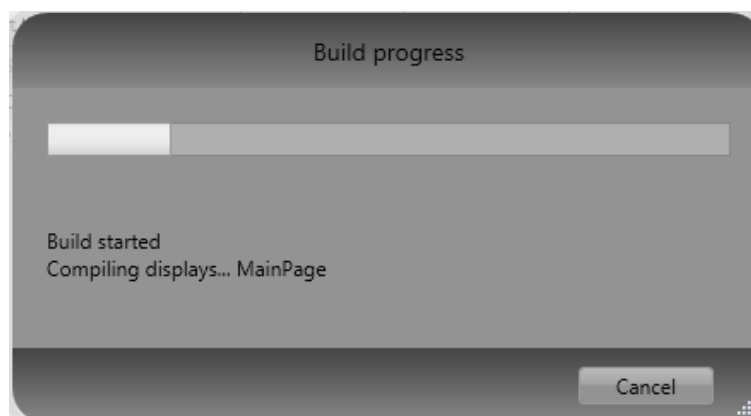



Figura 4-24. Progresso da Compilação

A Figura 4-25 ilustra o resultado da compilação e os erros, se existirem.

Figura 4-25. Resultado da Compilação

Teste do Projeto

A Figura 4-26 mostra os passos descritos para testar o projeto.

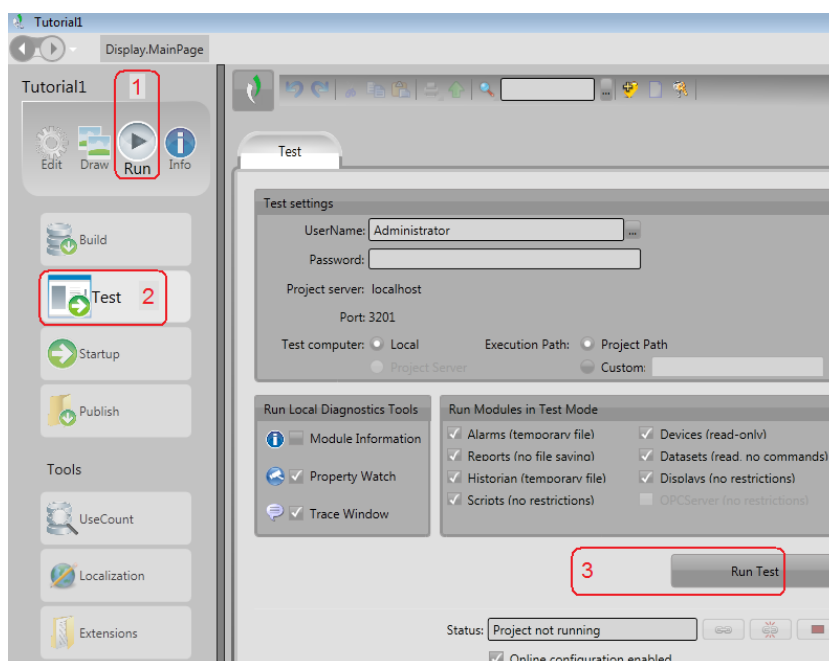



Figura 4-26. Teste do Projeto

1. Selecionar o menu Run
2. Selecionar a opção Test, que executa o projeto em modo teste. É possível configurar o teste, as ferramentas de diagnósticos e os módulos a serem testados
3. Iniciar o teste com as configurações definidas, confirmando através do botão Run Test

A Figura 4-27 mostra o resultado da execução do teste do projeto.

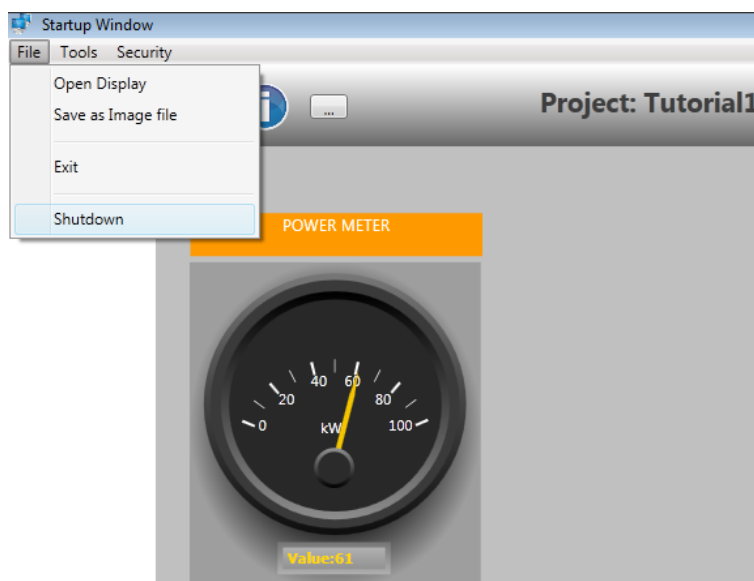



Figura 4-27. Execução do Teste do Projeto

Para finalizar o teste do projeto o usuário pode abrir o menu File e selecionar a opção Shutdown conforme representado na Figura 4-27.

Inicialização do Projeto

A Figura 4-28 mostra a sequência para inicialização do projeto.

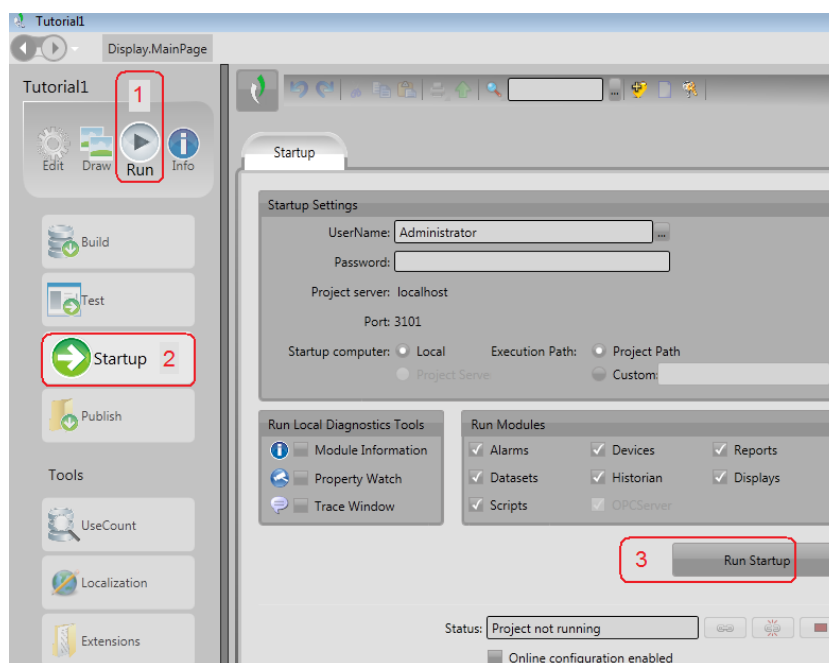



Figura 4-28. Execução da Inicialização do Projeto

1. Selecionar o menu Run
2. Selecionar a opção Startup, que permite configurar a inicialização, as ferramentas de diagnósticos e os módulos a serem inicializados
3. Iniciar a execução do projeto com as configurações definidas, através do botão Run Startup

Juntamente com a visualização da tela principal do projeto será aberta uma janela mostrando o status e as mensagens de inicialização conforme pode ser observado na Figura 4-29.

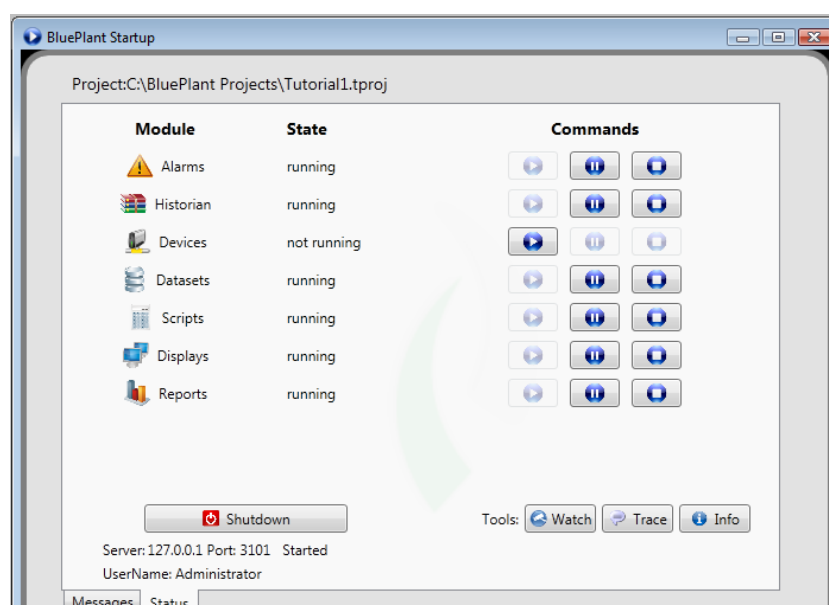



Figura 4-29. Status da Inicialização do Projeto

Publicação do Projeto

Quando o projeto está pronto para ser executado no campo, deve ser utilizado o recurso de publicação para configurar as opções de redundância (se aplicável) e para criar uma cópia do projeto na modalidade somente leitura com versão controlada, ou seja, uma cópia do projeto para a execução em campo. A extensão do arquivo gerado por ocasião da publicação é ".teng". A Figura 4-30 mostra a sequência para publicação do projeto.

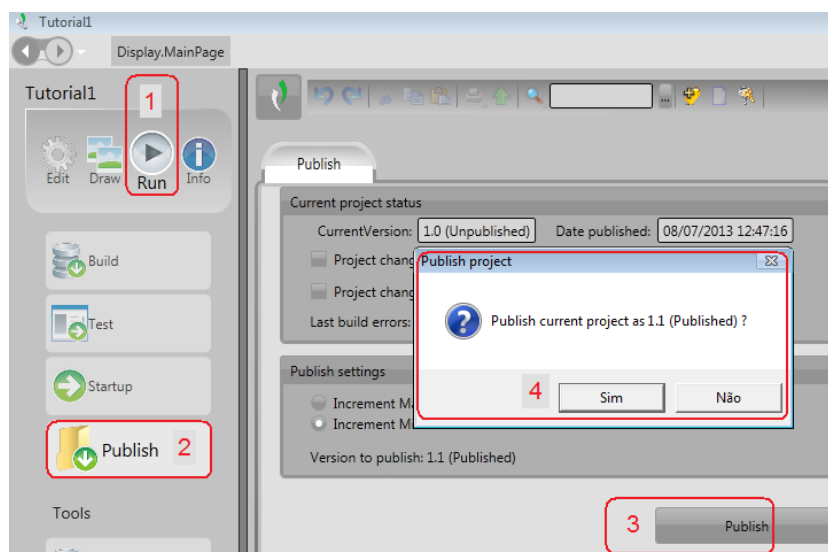



Figura 4-30. Publicação do Projeto

1. Selecionar menu Run
2. Acessar a opção Publish, que permite acessar as configurações da publicação do projeto
3. Clicar no botão Publish, que dispara a publicação do projeto com as configurações definidas
4. Janela de confirmação para publicação do projeto

Finalizada a publicação, uma janela similar à mostrada na Figura 4-31 será exibida indicando o caminho de publicação do projeto.

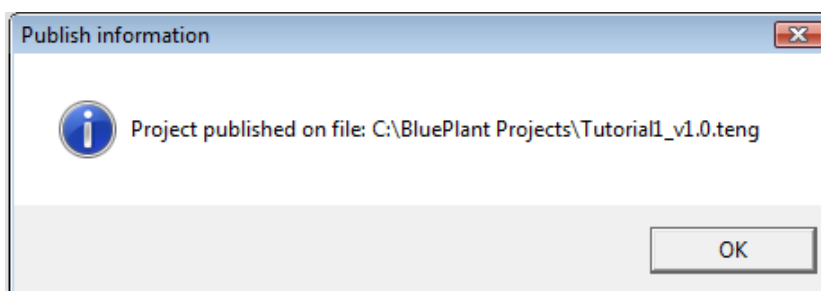



Figura 4-31. Caminho da Publicação do Projeto

5. Menu Principal do BluePlant

Cada sistema de supervisão gerencia certo número de objetos também denominados entidades que descrevem as variáveis de processo controladas e os elementos habituais do controle. A atividade de configuração de um supervisório compreende usualmente duas etapas:

- Definir cada variável de processo na base de dados
- Definir sinóticos, gráficos e relatórios

No sistema existem variáveis simples, primitivas e compostas, estas últimas formadas a partir das primeiras. Nesse contexto o nome da variável recebe o nome de Tag.

Neste capítulo são descritos os quatro menus básicos do BluePlant: Edit, Draw, Run e Info, os quais contêm as ferramentas necessárias para o desenvolvimento de um projeto de um sistema de supervisão contemplando as etapas típicas indicadas anteriormente. Esses quatro menus são descritos nas seções Edição da Aplicação, Diagramação da Aplicação, Execução da Aplicação e Informações da Aplicação respectivamente.

Edição da Aplicação

O menu Edit permite acessar as ferramentas para a edição de um projeto no BluePlant. A Figura 5-1 ilustra a seleção desse menu.

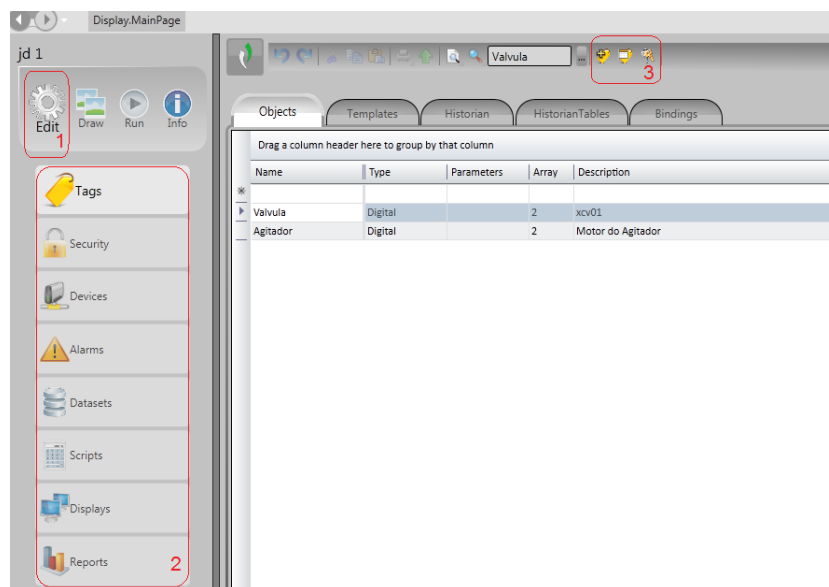



Figura 5-1. Menu Edit

Os itens que compõem o menu Edit estão numerados na figura anterior e são descritos na sequência:

1. Barra de seleção do menu de edição do projeto
2. Barra de ferramentas do menu de edição, contendo os recursos para a configuração dos seguintes elementos: tags, usuários e suas políticas de segurança, dispositivos e seus protocolos de comunicação, alarmes, banco de dados, linguagem Script, displays e relatórios
3. Atalhos na barra de ferramentas superior, incluindo as definições para criação de um novo tag, as propriedades das tags e as configurações de usuários bem como seu acesso aos elementos da aplicação

Diagramação da Aplicação

No menu Draw é possível criar as telas e símbolos que compõem a aplicação. A Figura 5-2 ilustra a inclusão de símbolos na tela principal (MainPage) com o auxilio das barras de ferramenta associadas.

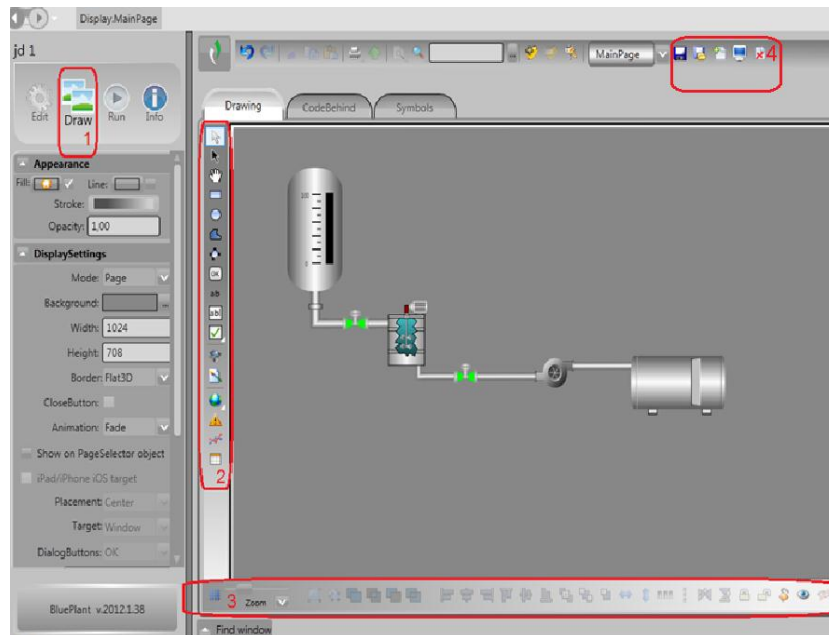



Figura 5-2. Menu Draw

Os itens que compõem o menu Draw estão numerados na figura anterior e são descritos na sequência:

1. Barra de seleção do menu de edição do projeto
2. Barra de ferramentas vertical usada para incluir, manipular e configurar os elementos que vão compor as telas da aplicação
3. Barra de ferramentas horizontal localizada abaixo da guia da tela, a qual contém comandos para agrupar, combinar, alinhar, bloquear o(s) componente(s) selecionado(s), entre outros
4. Barra de ferramentas TopToolBar localizada na parte superior da tela principal do menu Draw, a qual possui ícones com as seguintes funções: criar novo(s) display(s), apagar display(s), salvar display(s) e visualizar o(s) display(s) já existente(s) no projeto

Execução da Aplicação

O menu Run permite o acesso a todas as funcionalidades de execução do projeto. Os itens de menu disponíveis neste ambiente são: Buid, Test, Startup, Publish e o módulo Tools. A Figura 5-3 ilustra esse menu.

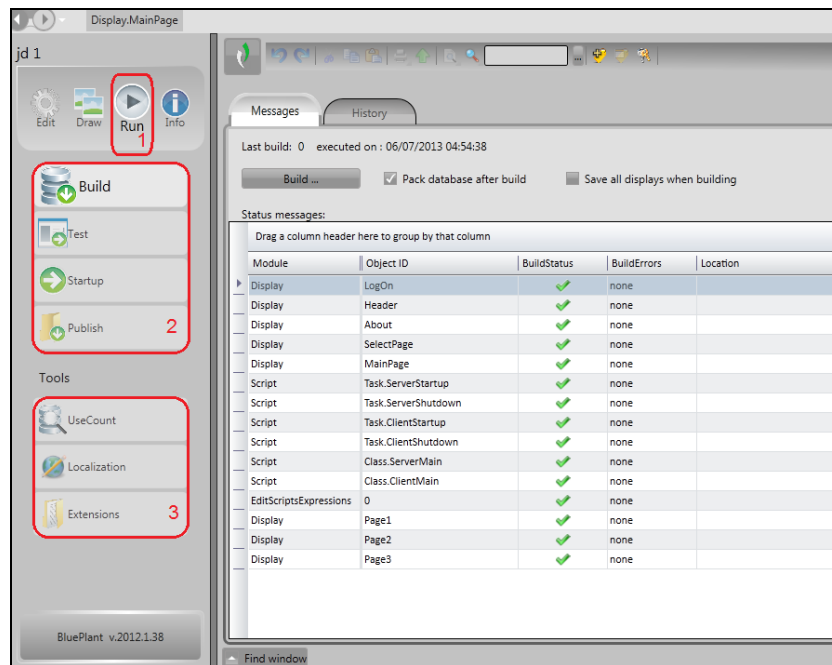



Figura 5-3. Menu Run

Os itens que compõem o menu Run estão numerados na figura anterior e são descritos na sequência:

1. Barra de seleção do menu de execução do projeto
2. Funcionalidades da execução do projeto
    - Comando Build, o qual prepara um projeto para execução e o salva no Histórico de Compilação do Projeto. Ele está dividido em Compilação de Mensagens e Compilação de Histórico. A Compilação de Mensagens exibe as informações de status, erros, localização e módulo. A Compilação de Histórico exibe as informações de compilação, erros, data da execução e usuários
    - Comando Test, contemplando as configurações associadas ao teste do projeto
    - Comando Startup para inicializar as configurações do projeto
    - Comando Publish que permite acessar as configurações da publicação do projeto
3. Módulo Tools, incluindo as funcionalidades contagem de tags e objetos, via comandos "UseCount" e "CrossReference", configurações de tradução no Runtime (Localization) e ferramentas de execução e Add-ons (Extentions)

Informações da Aplicação

O ambiente Info permite o acesso às informações do projeto. A Figura 5-4 ilustra esse menu.

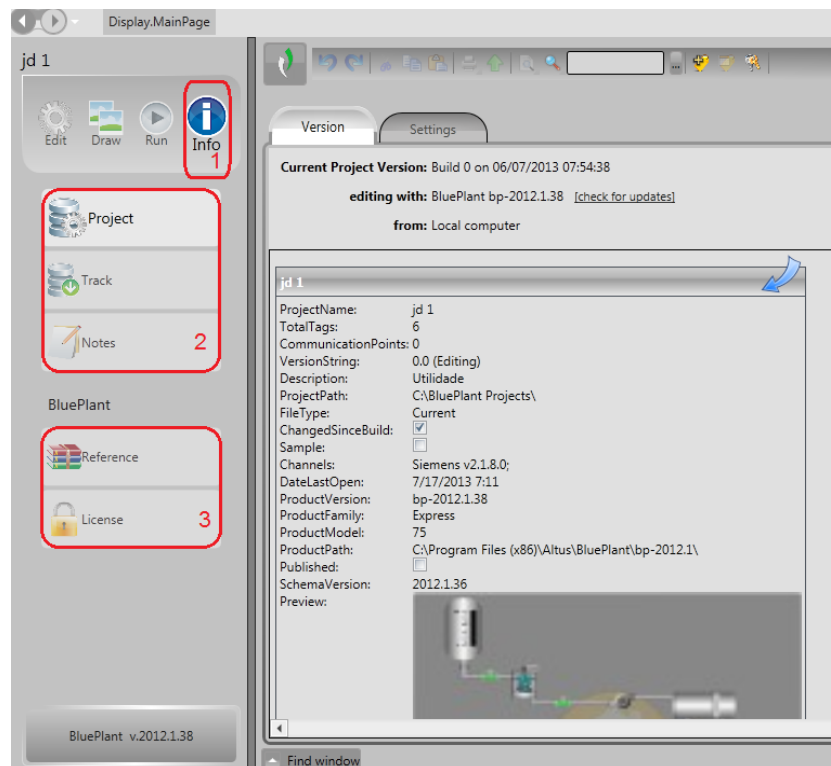



Figura 5-4. Menu Info

Os itens que compõem o menu Info estão numerados na figura anterior e são descritos na sequência:

1. Barra de seleção do menu de informações do projeto
2. Informações da aplicação, incluindo detalhamento das configurações do projeto (Project), rastreamento das alterações efetuadas (Track) e notas do usuário (Notes)
3. Informações do aplicativo onde é possível acessar a ajuda do software e obter informações sobre a licença de uso da ferramenta

6. Componentes do BluePlant

Este capítulo detalha as ferramentas de configuração, runtime e aplicativos do BluePlant. A seguinte estruturação foi considerada:

- Ferramentas de Configuração: menus Edit, Draw, Run e Info
- Ferramentas de runtime: ferramentas da aplicação, objetos runtime e linhas de comando
- Aplicativos BluePlant: telas em outros sistemas operacionais e serviço no Windows

Menu Edit

O ambiente de edição fornece acesso a todas as funcionalidades necessárias para configurar o projeto e é composto pelos itens listados na tabela a seguir.

Funcionalidade	Representação Gráfica
Edição de Tags	Tags
Definições de Segurança	Security
Configuração de Dispositivos	Devices
Planejamento dos Alarmes	Alarms
Banco de Dados	Datasets
Linguagem Script	Scripts
Projeto das Telas	Displays
Configuração de Relatórios	Reports

Tabela 6-1. Funcionalidades do Menu de Edit

Nas próximas seções serão detalhadas essas funcionalidades.

Editando Tags

O menu Edit Tags configura o banco de dados de tags em tempo real.

Edição de ObjetosTag

Use os tags (e suas propriedades pré-definidas) incluídos na lista a seguir para configurar um banco de dados em tempo real. Tipos disponíveis podem ser estendidos e novos tipos criados via tabela EditTagsUserTypes.

Nota:

"Tag" no contexto de uma configuração de projeto refere-se a uma variável de processo. A Figura 6-1 ilustra a edição de objetos Tags.

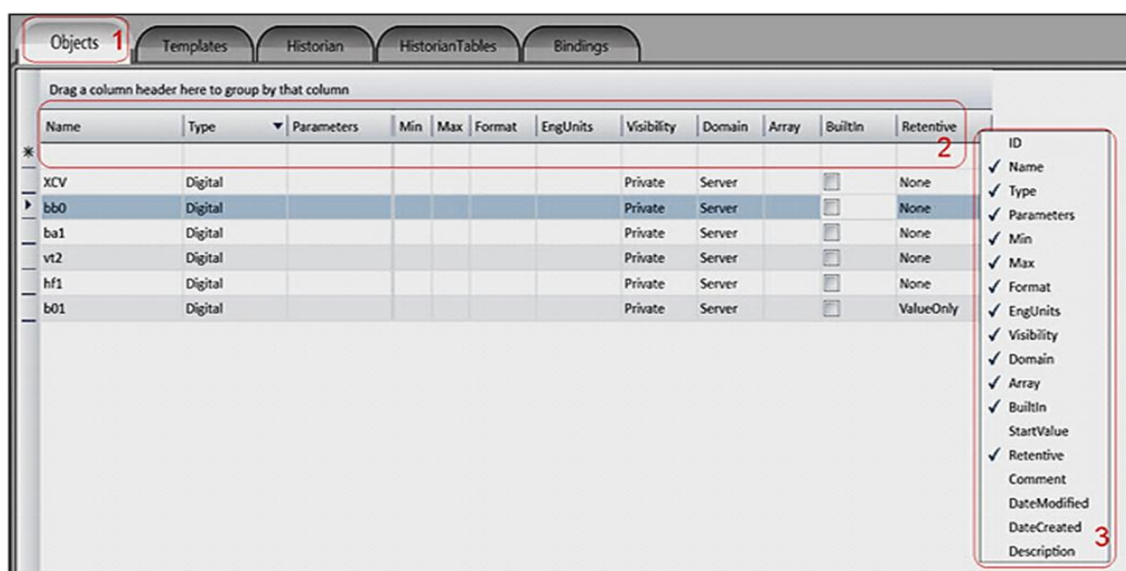



Figura 6-1. Edição de objetos do Tag

Os itens que compõem o menu de edição de objetos estão numerados na figura anterior e são descritos na sequência:

1. Selecionar a guia Objects;
2. Digitar os dados relativos ao novo Tag da aplicação na linha marcada com um asterisco (*);
3. Selecionar as opções de configuração dos tags. Clicar com o botão direito do mouse no cabeçalho de uma coluna e marcar os itens para exibição

Na sequência são explicados cada um dos itens do menu de edição de objeto conforme ilustrado na Figura 6-1.

Name

Define o nome do tag. Nesta coluna é possível criar ou editar tags.

Type

Estabelece o tipo do tag. Os tipos disponíveis são:

Tipo	Descrição
Digital	True ou False
AnalogInt	Inteiro
AnalogDecimal	Decimal
AnalogDouble	Ponto Flutuante
Text	Texto
Timer	Tempo
Counter	Contagem
Reference	Objeto referência apontando para outro objeto
DateTime	Data e hora
UserTypes	Tipos definidos pelo usuário

Tabela 6-2. Tipos de Disponíveis para Tags

Nota:

No caso de um tipo Reference, um objeto de referência deve ser inicializado para apontar para outro objeto. Isso normalmente é feito usando a seguinte sintaxe no corpo do script:

```
@Tag.Reference1.Link = @Tag.TagName.GetName();     (VB)
@Tag.Reference1.Link = @Tag.TagName.GetName();     (C#)
```

Parameters

Define parâmetros do tag de acordo com o tipo: banda morta para tags analógicos e ajustes para tags do tipo contador e temporizador. Quando criando "Tag de referência" (ou PONTEIROS) define o tag de referência de destino na coluna "Tipo de referência".

Min

Configura o valor mínimo admissível para o tag.

Max

Configura o valor máximo admissível para o tag.

Eng Units

Define a unidade de engenharia base para o tag.

Format

Especifica o formato de exibição de um valor. Para formatos numéricos válidos consulte o tópico Formatos Numéricos Padronizados. Exemplo: N1 (número com uma casa decimal). Para formatos de data e hora válidos consulte o tópico Formatos de Data e Hora Padronizados. Exemplo: d (data abreviada). Para uma discussão mais aprofundada sobre caracteres de formato consulte o tópico Tipos de Formatação. Exemplo de formatos numéricos:

Especificação	Descrição
N0	Número sem casas decimais
N3	Número com 3 casas decimais
X	Hexadecimal (suportado apenas em tipos inteiros)
C	Moeda

Tabela 6-3. Exemplo de Formatos Numéricos

Exemplo de formatos DateTime:

Especificação	Descrição
T (somente)	Padrão de tempo no formato longo (equivalente a "HH:mm:ss"))
d (somente)	Padrão de data abreviada (equivalente a "M/d/yyyy")(mês/dia/ano) (EUA)
dd	Representa o dia do mês como um número de 01 a 31
ddd	Representa o nome abreviado do dia da semana
dddd	Representa o nome completo do dia da semana
MM	Representa o mês como um número de 01 a 12
MMM	Representa o nome abreviado do mês
yy	Representa o ano como um número de dois dígitos
yyyy	Representa o ano como um número de quatro dígitos
hh	Representa a hora como um número de 01 a 12
HH	Representa a hora como um número de 00 a 23
mm	Representa o minuto como um número de 00 a 59
ss	Representa os segundos como um número de 00 a 59
fff	Representa os milissegundos como um número de 000 a 999
tt	Representa o marcador AM/PM

Tabela 6-4. Exemplo de Formatos DateTime

Exemplo: padrão de tempo no formato longo

Formato = HH:mm:ss (hora;minuto;segundo)

Visibility

Define a visibilidade do valor do Tag no servidor OPC para projetos remotos. As seguintes opções de configuração estão disponíveis:

- Privado: define um Tag visível somente no projeto local e par redundante
- Protegido: define um Tag do tipo somente leitura visível no servidor OPC DA para projetos remotos e clientes OPC DA
- Público: define um Tag visível no servidor OPC DA para projetos remotos e clientes OPC DA

Domain

Define o valor do Tag para o projeto inteiro ou um valor específico para cada tela cliente. As seguintes opções de configuração estão disponíveis:

- Servidor: o valor do Tag é consistido em todo o projeto e em todos os clientes
- Cliente: o valor do Tag é local para cada computador remoto executando uma visualização do cliente (Web ou visor)

Nota:

Tags do tipo cliente não devem ser usados em módulos do servidor como Device, Alarm, Historian e ServerScripts, uma vez que seus valores são locais (restritos ao computador em execução) e não são propagados para clientes remotos.

A maioria dos tags em um projeto é definida como "Servidor". tags "Locais", por sua vez, permitem valores diferentes em cada computador cliente. No entanto, pode-se usar tags "Locais" em dados temporários específicos para computadores-cliente individuais. A situação mais comum para o uso de tags "Locais" é quando dados temporários são necessários para gerenciar a Interface do usuário na tela. tags "Locais" permitem ainda valores diferentes em cada computador cliente.

Array

Quando não definido (em branco) o Tag não é do tipo matriz. Ao definir um Array com o valor N uma matriz de 0 até N posições é criada. Por exemplo, ao criar um Tag Array de tamanho "5", a matriz é criada a partir do Tag[0] ao Tag[5], significando que 6 elementos são criados. Esse método contempla dois estilos de programação: a indexação a partir do zero e a contagem a partir do 1.

Nota:

Em relação à extensão de Array por níveis de contas, as versões Lite e Express estão limitadas a matrizes unidimensionais. A versão Enterprise permite criar matrizes com até três dimensões.

Exemplo 1

```
Campo Name: TagAnalog
Campo Array: 2
```

Cria 3 elementos: TagAnalog[0] TagAnalog[1] TagAnalog[2]

Sintaxe de acesso:

```
C#: TagAnalog[1]
VB.NET: TagAnalog(1)
```

Exemplo 2

```
Campo Name: Temp
Campo Array: 1,2
```

Cria 6 elementos: Temp[0][0] Temp[0][1] Temp[0][2] Temp[1][0] Temp[1][1] Temp[1][2]

Sintaxe de acesso:

```
C#: Temp[1,2]
VB.NET: Temp(1,2)
```

Startup Value

Estabelece o valor de inicialização do Tag. Quando deixado em branco nenhum valor de inicialização é aplicado.

Retentive

Especifica se as Propriedades do Tag são retidas quando do encerramento do aplicativo, sendo usadas como valores de inicialização na próxima execução. As seguintes opções de configuração estão disponíveis:

- Não-Retentivo: o valor do Tag não é salvo
- ValueOnly: o valor do Tag é salvo quando modificado e pode ser usado como valor de inicialização na próxima execução
- Propriedades: todas as propriedades do Tag são salvas quando este é modificado para serem usadas na próxima inicialização

Nota:

Durante a edição é necessário pressionar a tecla ENTER na célula da tabela para a confirmação das configurações efetuadas.

Edição de Tipos Definidos pelo Usuário

Um tag de tipo existente pode ser localizado no campo "Tipo Personalizado do Usuário".

Clique [Del] para apagar um tag de tipo estrutura existente e [New] para criar um novo tag de tipo estrutura.

Um tag de tipo do usuário pode ser usado da mesma forma que os tags incorporados.

Nota:

Nas versões Express e Lite os tags de tipo do usuário podem ser usados somente na lista principal de tags. A versão Enterprise permite a criação de um tipo do usuário com até quatro níveis.

Exemplo: ao criar um tipo do usuário PID com os seguintes membros: 'setpoint' e 'PV', pode-se criar um tag denominado 'loop' do tipo PID, sendo que a sintaxe para acesso ao seu valor é:

```
loop.setpoint e loop.PV
```

A Figura 6-2 ilustra a edição de tipos definidos pelo usuário.

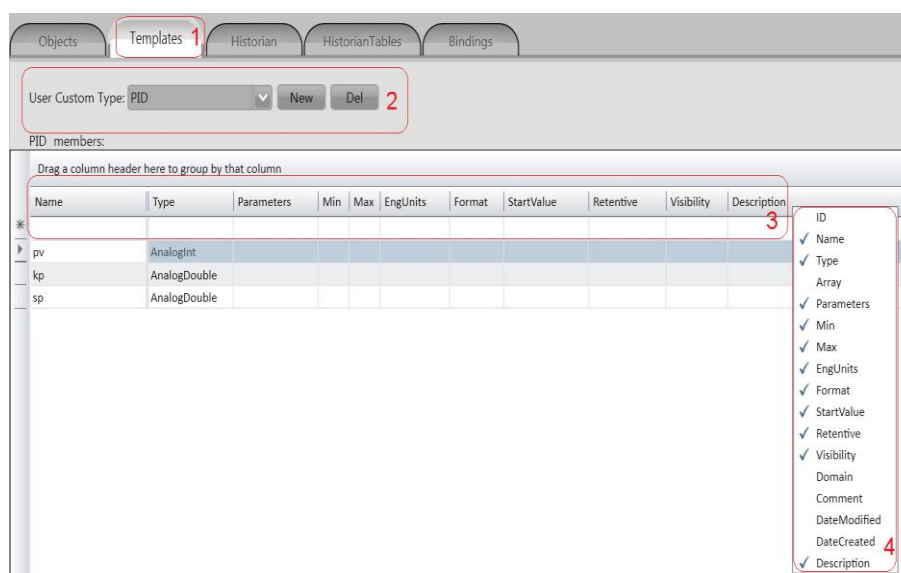



Figura 6-2. Edição de Templates Tag

Os itens que compõem o menu de edição de tipos estão numerados na figura anterior e são descritos na sequência:

1. Selecionar a opção Templates
2. Selecionar a opção User Custom Type clicando com o botão esquerdo do mouse em New para criar um novo elemento e em Del para excluir o elemento
3. Digitar os dados relativos ao novo tipo na linha marcada com um asterisco (*)
4. Selecionar as opções desejadas de configuração da coluna Tipo da tabela clicando com o botão direito do mouse no cabeçalho de uma coluna e marcando os itens para exibição

Na sequência são explicados cada um dos itens do menu de edição de templates conforme ilustrado na Figura 6-2.

Name

Define o nome do elemento. Nesta coluna é possível criar ou editar tipos do usuário.

Type

Define o tipo de tag. As opções de configuração de tipo do componente do tag são:

Tipo	**Descrição**
Digital	True ou False
AnalogInt	Inteiro
AnalogDecimal	Decimal
AnalogDouble	Ponto Flutuante
Text	Texto
Timer	Tempo
Counter	Contagem
Reference	Objeto referência apontando para outro objeto
DateTime	Data e hora
UserTypes	Tipos definidos pelo usuário

Tabela 6-5. Tipos de Dados Base para Gerar Novos Tipos do Usuário

Array

Quando não definido o tag não é do tipo matriz. Ao definir um array com o valor N, uma matriz de 0 até N posições é criada. Por exemplo, ao criar um tag array de tamanho cinco (5), a matriz é criada a partir do tag[0] ao tag[5], significando que 6 elementos são criados.

As versões Lite e Express estão limitadas a matrizes unidimensionais. A versão Enterprise permite criar matrizes com até três dimensões.

Reference

Configura tags do tipo referência. Este campo especifica o tipo de destino quando um tag reference é criado. Um tag de referência pode apontar - de forma dinâmica e em Runtime - para diferentes tags conforme os tipos definidos neste campo.

Min/Max

Define o valor mínimo e máximo do tag. Não se podem definir valores de Tag menores do que o valor mínimo ou maiores do que o valor máximo.

Eng Units

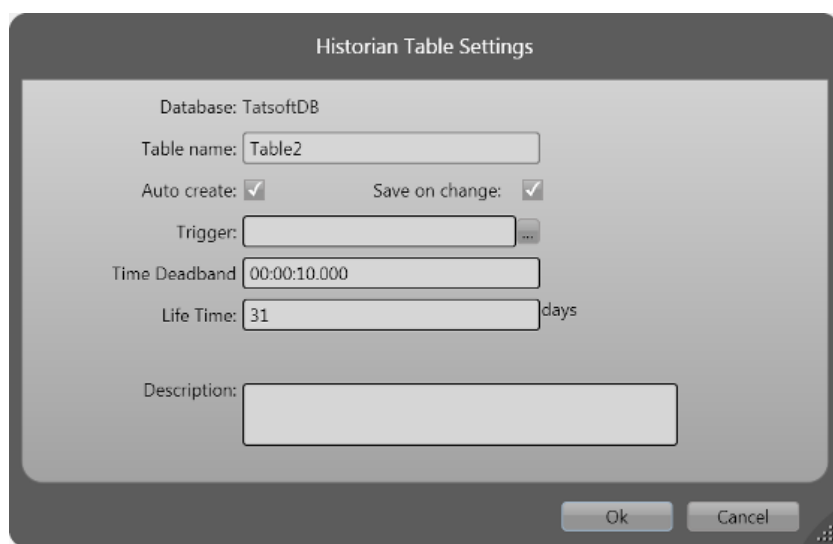
Define a unidade de engenharia base para o Tag.

Start Value

Estabelece o valor de inicialização do Tag. Quando deixado em branco nenhum valor de inicialização é aplicado.

Retentive

Especifica se as propriedades do Tag são retidas quando do encerramento do aplicativo, sendo usadas como valores de inicialização na próxima execução. As opções disponíveis são:

- Não retentivo: o valor do Tag não é salvo
- ValueOnly: o valor do Tag é salvo quando modificado para ser usado como valor de inicialização na próxima execução

Todas as propriedades do tag são salvas quando este é modificado para serem usadas na próxima inicialização.

Ao aplicar propriedades retentivas, as modificações são salvas no banco de dados<project>.retentiv. O uso de configurações retentivas para tags cujos valores mudam rapidamente não é recomendado uma vez que isso pode piorar o desempenho da execução. Este é o caso, por exemplo, das variáveis do processo que são consideradas críticas e/ou estão associadas a condições de segurança.

Edição de Historiador

É possível configurar um banco de dados de Historiador para registrar alterações nos tags. Para selecionar uma Tabela de Historiador existente pesquise no campo "Tabelas de Historiador".

Clique Del para apagar uma Tabela de Historiador existente ou clique New para configurar uma nova Tabela de Historiador. A Figura 6-3 mostra as configurações da Tabela do Historiador.

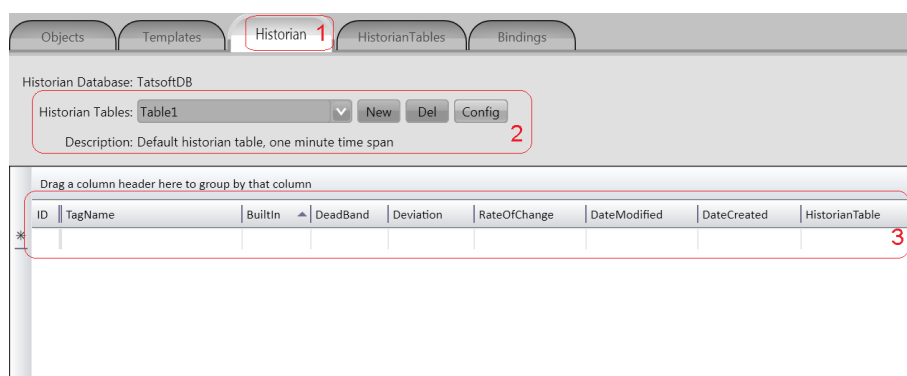



Figura 6-3. Configurações da Tabela do Historiador

O banco de dados onde os tags são salvos é definido em EditDatasetDBs no DB chamado "TagHistorian".

A Figura 6 4 ilustra as configurações associadas ao historiador de tags.

Figura 6-4. Edição de Historiador

Os itens que compõem a tabela de edição de historiador estão numerados na figura anterior e são descritos na sequência:

1. Selecionar a guia Historian clicando com botão esquerdo do mouse
2. Na opção Historian Tables clicar com o botão esquerdo do mouse em New para criar uma nova tabela e em Del para excluir uma tabela. O botão Config permite acesso às configurações da tabela do historiador mostradas na Figura 6-3
3. Digitar os dados relativos ao Tag cujas alterações desejam-se registrar na linha marcada com um asterisco (*)

Na sequência são explicados cada um dos itens do menu de edição de historiador conforme ilustrado na Figura 6-4.

TagName

Indica o nome do tag. Nesta coluna é possível inserir ou remover tags do historiador.

DeadBand

Configura a banda morta do Historiador. O evento de salvamento no registro é disparado somente se o tag tem seu valor alterado acima da banda morta especificada.

Exemplos:

Banda morta configurada 10. Valor atual 20. Todos os novos valores (10 ou mais) maiores do que o valor anterior são adicionados ao registro do historiador.

Valor alterado para 35 (diferença em relação ao valor anterior que era 20:15). Resultado: valor adicionado ao registro do historiador.

Valor alterado para 50 (diferença em relação ao valor anterior que era 35: 15). Resultado: valor adicionado ao registro do historiador.

Valor alterado para 55 (diferença em relação ao valor anterior que era 50: 5). Resultado: valor não é adicionado ao registro do historiador.

Deviation

Se o SaveOnChange estiver definido nas configurações da tabela e o Tag sofrer uma alteração maior do que o parâmetro Deviation (valor anterior e atual > Deviation), o valor será salvo imediatamente (não aguarda o próximo TimeSpan).

RateOfChange

Se o Tag RateOfChange (unidades de engenharia por segundo) for maior que o parâmetro RateOfChange especificado e SaveOnChange estiver habilitado, a tabela será salva imediatamente (não aguarda o próximo TimeSpan).

HistorianTable

Define a tabela do banco de dados onde os tags "Historiador" são salvos. Clique Config para editar as configurações da tabela do historiador conforme mostrado na Figura 6-3. As opções de configuração são:

- TableName: nome da tabela do banco de dados
- TimeSpan: tempo mínimo entre registros
- Trigger: a tabela é salva a cada alteração no disparo
- AutoCreate: cria automaticamente uma tabela quando esta não é encontrada no banco de dados
- LifeTime: quando definido (maior que "0") exclui registros mais antigos do que o ciclo de vida

- Description: descrição definida pelo usuário para fins de documentação

Associação de Tags

É possível configurar dados de associação entre tags ou propriedades, de forma que quando os dados alteram seu valor os elementos a eles associados alteram-se automaticamente. A associação de dados pode também significar que se outra representação dos dados em um elemento alterar-se, então os dados subjacentes serão também atualizados automaticamente. A Figura 6-5 mostra essa seleção.

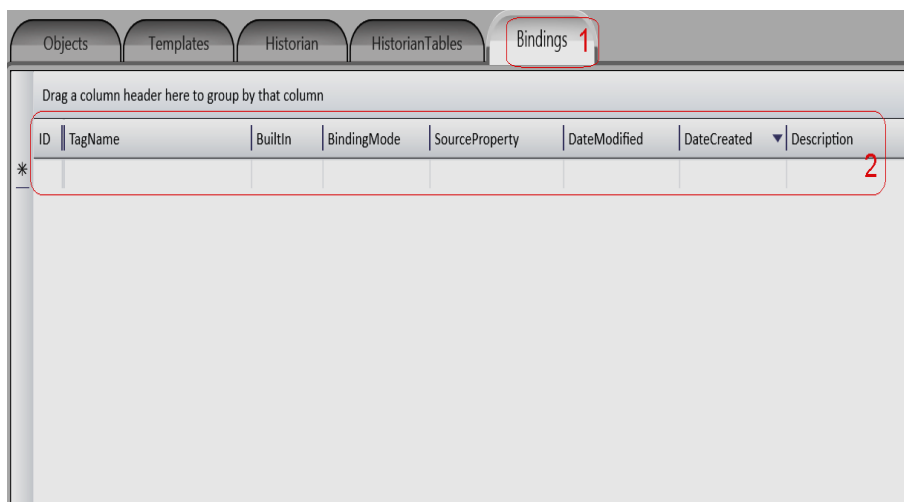



Figura 6-5. Edição de Associação de Tags

Os itens que compõem o menu de edição de associações estão numerados na figura anterior e são descritos na sequência:

1. Selecionar a guia Bindings clicando com botão esquerdo do mouse
2. Digitar os dados relativos ao Tag na linha marcada com um asterisco (*)

Na sequência são explicados os itens do menu de edição de associações conforme ilustrado na Figura 6-5.

Name

Especifica o nome do tag associado.

BindingMode

Define o modo de associação com as seguintes opções de configuração:

- OneTime (na inicialização)
- OneWay (nome do tag altera-se quando a sourceproperty alterar-se)
- TwoWay (qualquer alteração no nome do Tag ou na sourceproperty altera o campo oposto)
- OneWayToSource (sourceproperty altera-se quando o nome do tag alterar-se)

Source Property

Estabelece a propriedade da fonte de dados.

Editando as configurações de segurança

O menu Edit Security define as permissões do usuário e níveis de acesso ao projeto.

Edição de Usuários

Define os usuários do projeto e as permissões relacionadas. As Ids dos níveis Administrador e Hóspede apresentam atributos incorporados. Algumas permissões de usuário permanecem ativas independentemente da alteração de nome ou de permissões, conforme segue:

Usuário Administrator (ID:2): a ID do Administrator permanece a mesma mesmo se o nome do usuário ou as permissões forem alteradas. O Administrator é o único usuário que pode apagar ou bloquear usuários e pode definir senhas para as interfaces do Banco de Dados.

Usuário Guest (ID:0): esta ID é usada para usuários de login anônimo. Usuários hóspedes não possuem senhas atribuídas. Entretanto, suas permissões podem ser alteradas. Quando o sistema é inicializado sem Usuário(s) definido (ou se é um login anônimo) se aplicam as permissões de segurança definidas para o usuário hóspede.

A Figura 6-6 mostra a edição dos perfis dos usuários no âmbito da segurança.

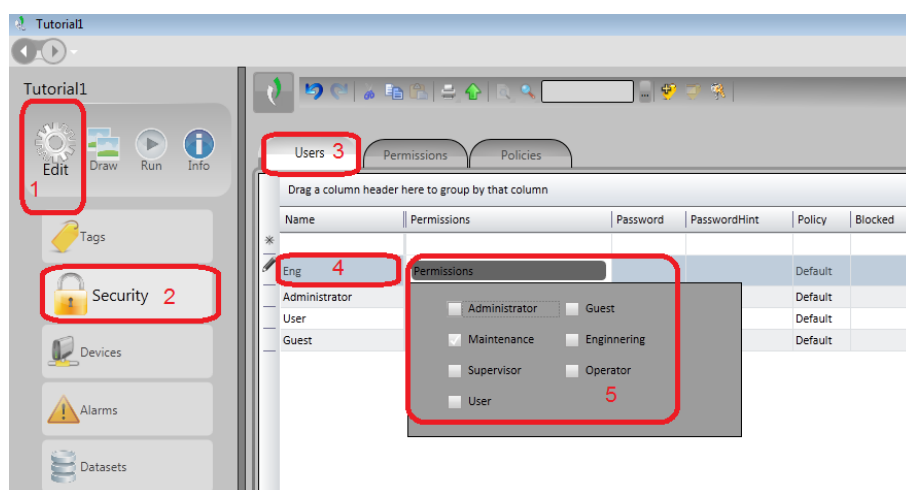



Figura 6-6. Edição de Perfis de Usuários

Os passos para edição/inclusão de perfis de usuários estão numerados na Figura 6-6 e são descritos a seguir:

1. Selecionar o menu Edit
2. Acessar o item Security
3. Clicar na guia Users para acesso aos usuários do sistema
4. Digitar os dados relativos aos usuários da aplicação na linha marcada com um asterisco (*)
5. Selecionar as permissões disponíveis que o usuário adicionado ou em edição pode ter no sistema

ATENÇÃO É possível configurar uma senha e uma dica da senha para cada usuário.

Edição de Permissões

Define a configuração e os privilégios de execução para cada grupo de permissão. A Figura 6 7 ilustra essa seleção.

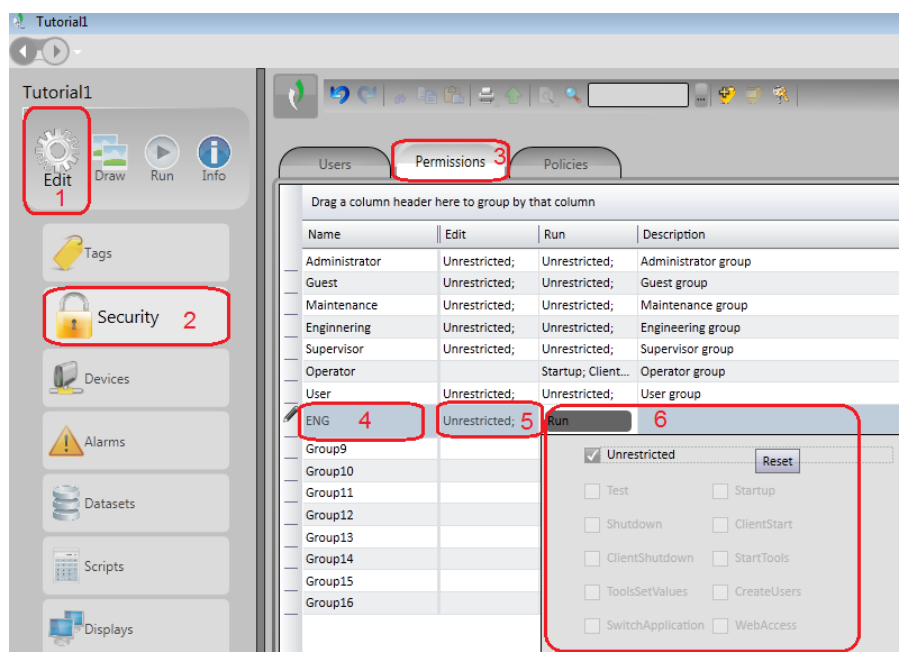



Figura 6-7. Edição de Permissões de Segurança

Os passos para editar as permissões existentes estão ilustrados na Figura 6-7 e descritos a seguir:

1. Selecionar o menu Edit
2. Acessar o item Security
3. Clicar na guia Permissions para acesso às permissões do sistema
4. Digitar os dados relativos às permissões dos usuários na linha marcada com um asterisco (*)
5. Configurar as permissões de edição na coluna Edit
6. Configurar as permissões do Runtime na coluna Run

ATENÇÃO:
A configuração das permissões de edição, serve para que os usuários configurados com esta permissão possam realizar determinados tipos de modificações no projeto. As configurações das permissões do Runtime servem para que os usuários configurados com esta permissão possam realizar determinadas ações durante a execução do projeto.

Name

Estabelece o nome do grupo de permissão.

Edit

Define os privilégios de grupos de permissão para edição e desenho. As opções disponíveis são:

- Unrestricted (irrestrito)
- EditTags (Edição de tags)
- Security (Segurança)
- Scripts (Scripts)
- Datasets (Banco de dados)
- Reports (Relatório)
- Publish (Publicação)
- Notes (Notas)
- Historian(Historiador)
- Alarms (Alarmes)

- Devices (Dispositivos)
- Displays (Telas)
- Startup (inicialização)
- Settings (Configurações)
- CreateTags (Criação de novas tags)

Run

Define os privilégios de Execução de Grupos de Permissão (Runtime). As opções disponíveis são:

- Unrestricted (irrestrito)
- Test (teste)
- Startup (inicialização)
- Shutdown (parada)
- ClientStart (inicialização do cliente)
- ClientShutdown (parada do cliente)
- StartTools (ferramentas de inicialização)
- ToolsSetValues (valores de ajuste das ferramentas de inicialização)
- CreateUsers (criação de usuários)
- SwitchApplication (alternar aplicação)
- WebAccess (acesso via Web)

Editando Dispositivos

O BluePlant é fornecido com um driver OPC DA para coleta de informações de dispositivos remotos. Além do OPC, o BluePlant também oferece suporte a drivers de comunicação personalizados para acessar diretamente CPs, sistemas remotos de E/S, barramentos de campo padronizados, laços únicos e múltiplos, scanners, leitores de código de barras, dispositivos RFID e monitores digitais. Consulte o manual de suporte ao Dispositivo para mais informações sobre protocolo e programação de cada dispositivo.

ATENÇÃO: Para mais informações sobre as configurações dos dispositivos (canais, nós e pontos de comunicação) deve ser acessado através do botão Help na guia Channels.

Canais

A aba Channels configura os protocolos e os canais de comunicação. A Figura 6 8 ilustra essa seleção.

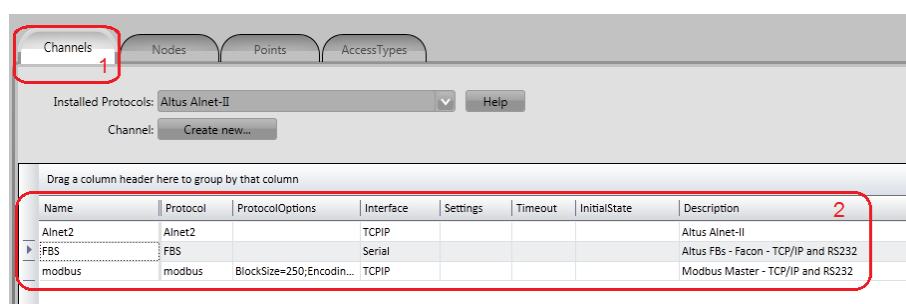



Figura 6-8. Edição de Canais de Comunicação

Os itens que compõem o menu de edição de canais de comunicação estão numerados na figura anterior e são descritos na sequência:

1. Selecionar a opção Channels
2. Observar que os dados relativos aos protocolos instalados aparecem na linha marcada com uma seta (►)

Na sequência são explicados os itens do menu de edição de canais de comunicação conforme ilustrado na Figura 6-8.

Name

Estabelece o nome do canal.

Protocol

Define o protocolo de comunicação em execução em um Canal específico.

ProtocolOptions

Contempla a configuração específica do protocolo.

Interface

Indica a interface de comunicação para o canal. As interfaces disponíveis dependem do protocolo escolhido. A Tabela 6-6 mostra as interfaces disponíveis para cada protocolo.

Protocolo	**Interfaces disponíveis**
Altus ALNET I	Serial e Multiserial
Altus ALNET II	TCP/IP
ControlLogix - Rockwell – Protocolo CIP	TCP/IP
Altus FBs - Facon	Serial, Multiserial e TCP/IP
MODBUS Master - TCP/IP e RS-232C	Serial, Multiserial e TCP/IP
MODBUS Slave - TCP/IP e RS-232C	Serial e TCP/IP
OPCXmlDA - OPC Xml/DA Client	OPC
Siemens – Protocolo S7	TCP/IP

Tabela 6-6. Protocolos e Interfaces

Settings

Inclui a definição das configurações da interface de comunicação. Cada tipo de interface possui ajustes específicos como, por exemplo, porta de comunicação, velocidade, bits de dados, bits de parada, paridade e sinais de controle. Um duplo clique na célula Settings permite acessar o menu suspenso com as configurações associadas à interface.

Timeout

Configurações de timeout para a interface de comunicação.

Criando um Canal de Comunicação

O botão Create new... pode ser utilizado para criar um novo canal. A Figura 6 9 ilustra essa seleção.

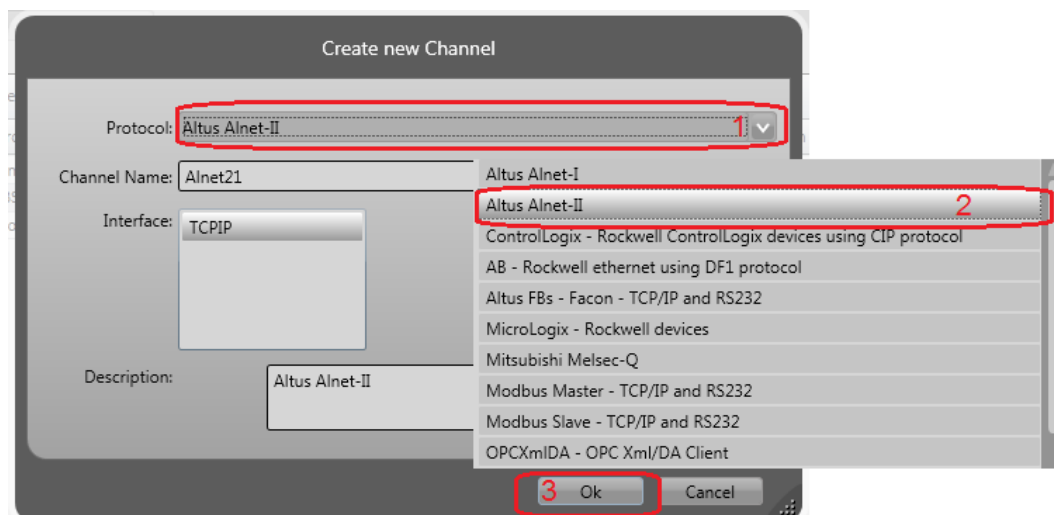



Figura 6-9. Edição de um Novo Canal de Comunicação

Na sequência será explicado o passo a passo para a criação de um novo canal de comunicação conforme exemplo ilustrado na Figura 6-9.

1. Selecionar a opção Protocol clicando com o botão esquerdo do mouse
2. Selecionar nas opções existentes, aquela desejada. Neste caso: Altus ALNET II
3. Finalizar clicando com o botão esquerdo do mouse no botão Ok

Nós

No âmbito da computação, um nó é um ponto ou terminal na rede, onde uma mensagem pode ser criada, recebida ou transmitida. No contexto dos sistemas de supervisão operando em rede, o nó pode ser um dispositivo eletrônico ativo que está ligado a uma rede, e é capaz de enviar, receber ou transmitir informações através de um canal de comunicação. A Figura 6-10 mostra o menu de edição de nós para um determinado canal de comunicação. Um canal de comunicação que, conforme indicado anteriormente, está associado a um protocolo específico e pode conter um ou mais nós, caracterizando os dispositivos conectados a esse canal.

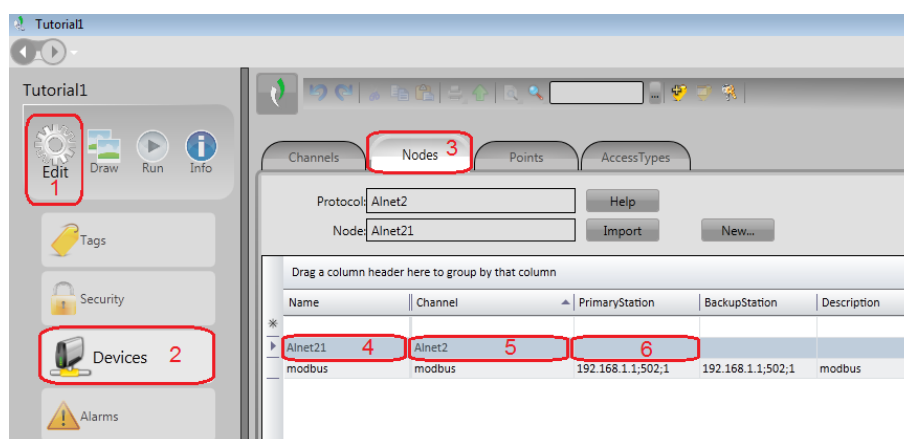



Figura 6-10. Edição do Nó para o Canal de Comunicação

Os passos para editar ou incluir os nós existentes estão ilustrado na Figura 6-10 e descritos a seguir:

1. Selecionar o menu Edit
2. Acessar o item Devices
3. Clicar na guia Nodes para acesso às configurações dos nós

4. Incluir ou editar o nome do nó na linha marcada com um asterisco (*)
5. Selecionar o Channel (protocolo) previamente adicionado no projeto
6. Indicar através da coluna PrimaryStation qual o endereço do equipamento que estará relacionado ao nó

Name

Indica o nome do nó.

Channel

Informa o canal de comunicação associado a este nó.

Primary Station

Referencia a estação do Nó. Define o Endereço IP, Porta e SlaveID. A sintaxe do campo da estação depende do protocolo. A figura a seguir mostra um exemplo de parametrização no caso do protocolo ALNET.

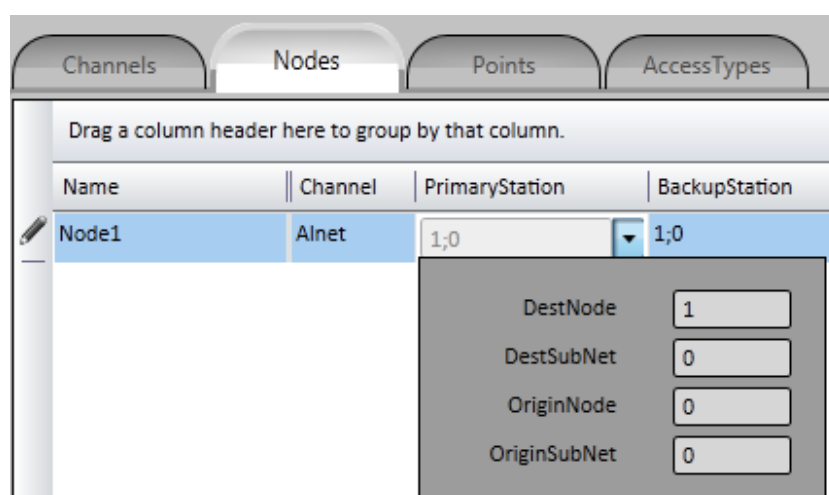



Figura 6-11. Parametrização da Estação Primária

Backup Station

Define o Endereço IP, Número de porta e SlaveID. Se este campo estiver definido, ao ocorrer uma falha de comunicação na estação principal, o sistema automaticamente tentará estabelecer uma comunicação com a estação de backup.

Points

Define os valores de aquisição de dados dos dispositivos de campo e mapeia os valores nos tags.

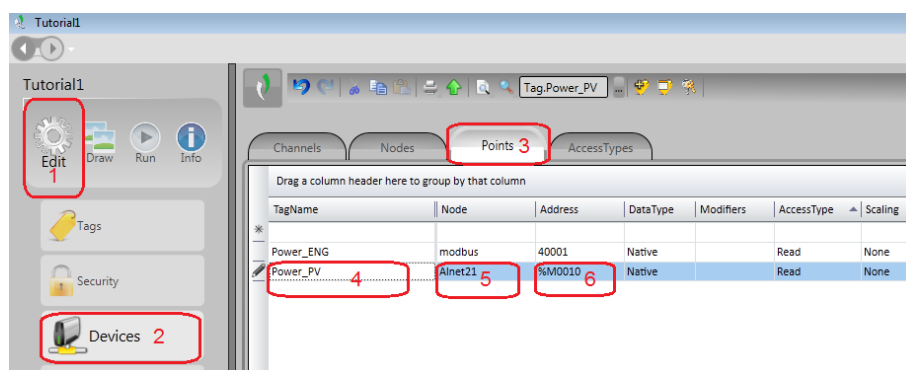



Figura 6-12. Edição de Pontos

Os passos para editar ou incluir os pontos de comunicação estão ilustrado na Figura 6-12 e descritos a seguir:

1. Selecionar o menu Edit
2. Acessar o item Devices
3. Clicar na guia Points para acesso às configurações dos pontos de comunicação
4. Incluir ou editar o nome da tag existente (selecionável) na linha marcada com um asterisco (*)
5. Selecionar o nó (dispositivo configurado) previamente adicionado no projeto
6. Indicar através da coluna Address qual o endereço do dado a ser recebido ou enviado do equipamento especificado

Name

Define o TagName a ser lido ou escrito no Dispositivo identificado.

Nodes

Define o Nó de comunicação associado ao ponto do dispositivo.

Address

Indica o endereço do ponto no dispositivo. A sintaxe do campo de endereço depende do protocolo. A Figura 6-13 ilustra um exemplo de parametrização do endereço (operando) no caso do protocolo ALNET.

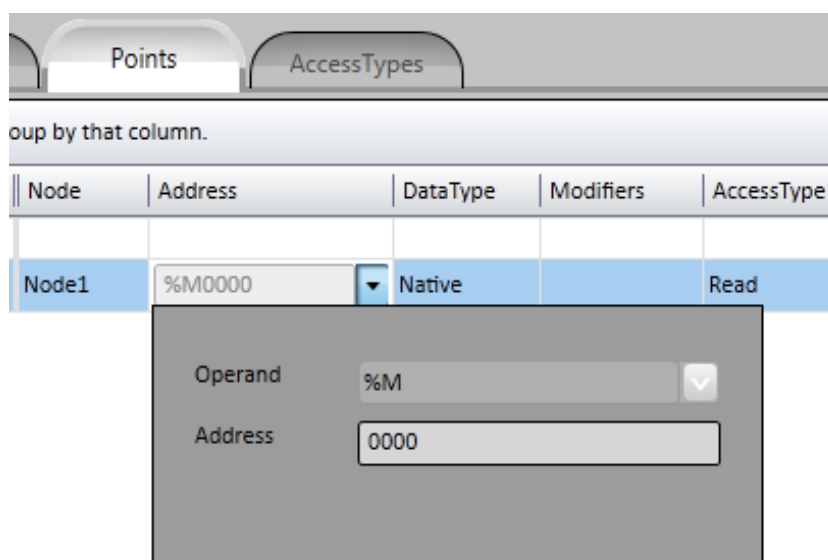



Figura 6-13. Exemplo de Parametrização do Endereço

Date Type

Define a conversão de dados aplicada aos dados de comunicação. A maioria dos protocolos deve utilizar a opção NATIVE. Neste caso, o protocolo automaticamente tratará da conversão de dados. Se um DataType diferente do NATIVE for selecionado, os protocolos padrão são sobrescritos. Os tipos de dados possíveis são:

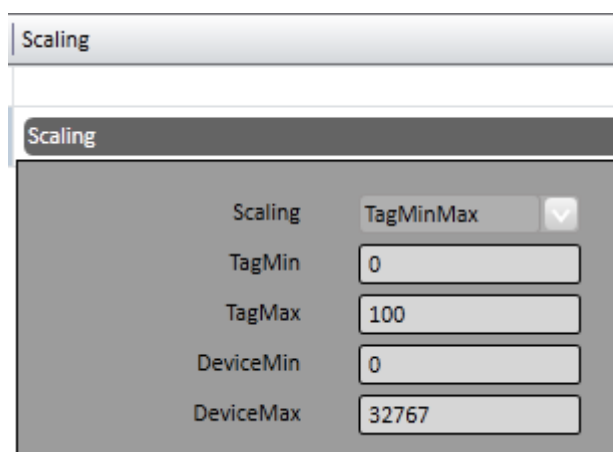
- Native (automático)
- Bit (binário – 1 bit)
- Byte (octeto – 8 bits)
- Char (menor unidade endereçável da máquina que pode conter caracteres básicos – 8 bits)
- Short (número inteiro com sinal - 16 bits)
- Dword (número inteiro – 16 bits)
- Integer (número sem ponto decimal)
- Long (número inteiro com sinal – 32 bits)
- ULong (número inteiro sem sinal – 32 bits)
- BCD (número decimal codificado como binário)
- LBCD (Long BCD)
- Single (número de ponto flutuante com precisão simples)
- Real (número real – ponto flutuante)
- ASCII (codificação de caracteres de sete bits baseada no alfabeto inglês)
- Unicode (representação e manipulação de texto)
- OPCDateTime (data e hora padrão OPC)
- Timer (codificação de tempo)
- Counter (codificação de contagem)
- Control (tipo de dado de controle)

Modifiers

Fornece a seleção de bit e outras configurações de dados de comunicação, podendo haver mudanças de acordo com o protocolo. Os seguintes campos podem ser definidos:

- Bit (unidade básica de informação binária – 0/1)
- ByteSwap (permuta de bytes)
- WordSwap (permuta de palavra)

- Stringlength (comprimento do string)

AccessType

Define o comportamento de leitura e escrita para cada ponto.

Scaling

Estabelece a conversão de escala aplicada usando os dados de comunicação. As configurações de conversão de escala são:

- None (nenhuma conversão)
- TagMinMax (escala baseada no valor mínimo e máximo do tag)
- Liner (escala linear)
- Equation (escala via equação)

A Figura 6-14 ilustra a parametrização de conversão de escala baseada na opção TagMinMax.

Figura 6-14. Conversão de Escala

Tipo de Acesso

Define as características do tipo de acesso comuns aos pontos do dispositivo. Ver Figura 6-15. Há três tipos de acesso padrão:

- ReadWrite (leitura e escrita)
- Write (escrita)
- Read (somente leitura)

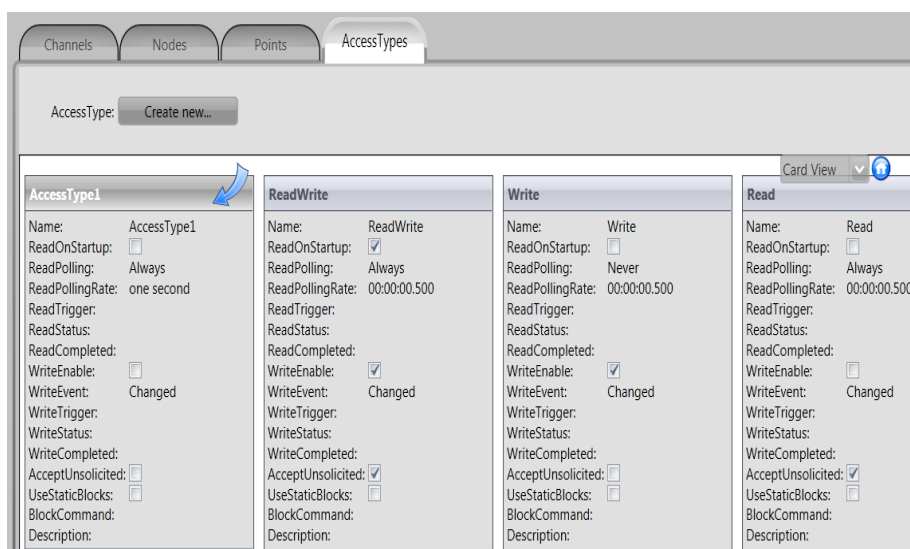



Figura 6-15. Tipo de Acesso

Clique em **Create new AccessType** para criar um novo AccessType.

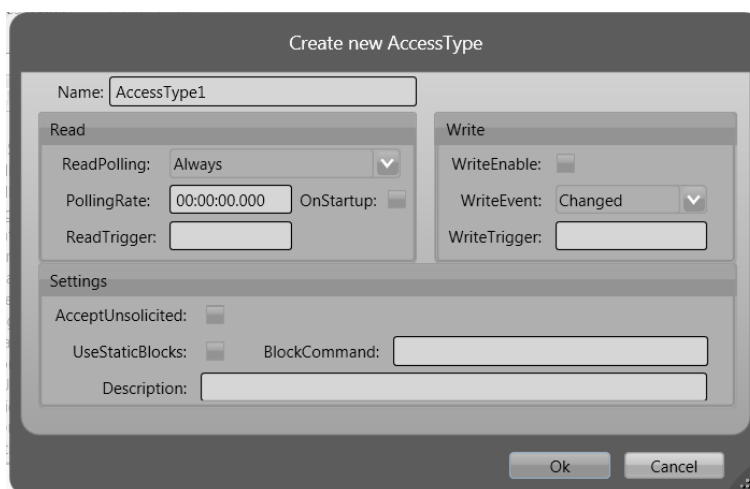



Figura 6-16. AccessType

Na sequência são explicados os principais itens do menu de tipo de acesso conforme ilustrado na Figura 6-16.

Name

Indica o nome do tipo de acesso.

ReadPooling

Define a leitura por pooling. Os atributos associados são:

- Selecionado = always = leitura por polling ativa
- Não selecionado = never= leitura por polling inativa

PoolingRate

Define a taxa de leitura por pooling para cada AccessType se ReadPooling estiver selecionado.

OnStartup

Define o ponto de leitura na inicialização. Os atributos associados são:

- Selecionado = true = habilita a leitura na inicialização
- Não selecionado = false = desabilita a leitura na inicialização

WriteEnable

Habilita ou desabilita a escrita quando ocorrer um evento. Os atributos associados são:

- Selecionado = true = habilita a escrita
- Não selecionado = false = desabilita a escrita

WriteEvent

Configura o tipo de acesso de escrita ao ponto. Os atributos associados são:

- Changed - escreve quando o Tag vinculado é alterado
- ChangedUp - escreve quando o Tag vinculado sofre um aumento de valor
- ChangedDown - escreve quando o Tag vinculado sofre um decréscimo de valor

AcceptUnsolicited

Define os atributos de aceite de mensagem não solicitada. Os atributos associados são:

- Selecionado - habilita mensagem não solicitada
- Não selecionado - desabilita mensagem não solicitada

Editando Alarmes

Um Alarme pode ser configurado através da associação de um Tag com um Grupo de Alarmes específico e um valor limite.

Para exibir um item de Alarme previamente configurado desenhe um objeto de alarme no ambiente de telas do seu projeto (Draw), inserindo o símbolo (Alarm Window) a partir da barra de ferramentas esquerda.

Os métodos para o reconhecimento de Alarmes incluem objetos de alarme na tela, propriedades do Tag, grupos de alarme ou propriedades do item alarme, bem como os campos "AckAll" (global) e "AckMostPriority".

Notas:

Reconhecer todos os alarmes: pode-se usar a propriedade <Alarm.AckAll> que reconhece todos os alarmes configurados em um projeto com <Edit.Alarms.Items>.

Reconhecer alarme único ou alarme de alta prioridade: a propriedade {Alarm.PriorityItem.UnAck} permite o reconhecimento do Alarme de alta prioridade configurado em <Edit.Alarms.Items> na coluna "Prioridade". Se este for o único Alarme ou se este for de alta prioridade ele será reconhecido, mas somente se estiver Ativo ou Normalizado.

Reconhecer alarme específico: para reconhecer um alarme específico, use a propriedade <Alarm.Items.IDxx.Unack>.

Grupos de Alarme

Define o comportamento do tratamento de alarme comum para um grupo de itens de Alarme.

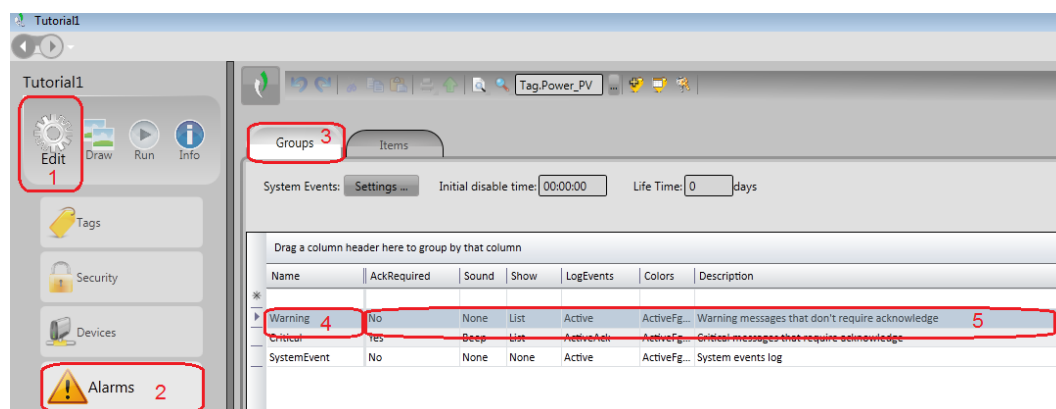



Figura 6-17. Grupos de Alarmes

Os passos para editar ou incluir os grupos de alarmes estão ilustrados na Figura 6-17 e descritos a seguir:

1. Selecionar o menu Edit
2. Acessar o item Alarms
3. Clicar na guia Groups para acesso às configurações dos grupos de alarmes
4. Incluir ou editar o nome do grupo de alarmes na linha marcada com um asterisco (*)
5. Selecionar as configurações do grupo especificadas nas colunas apresentadas na imagem acima

Name

Indica o nome definido pelo usuário para o grupo de alarmes. Os grupos de alarme "Warning", "Critical" e "SystemEvents" são incorporados e não podem ser excluídos, embora seus nomes e configurações possam ser alterados.

AckRequired

Configura as opções de reconhecimento para os alarmes do grupo. Os atributos associados são:

- Não = 0 - Pontos de alarme definidos no grupo não requerem reconhecimento
- Sim = 1 - Alarmes definidos no grupo requerem reconhecimento

Sound

Habilita ou desabilita os sons do alarme quando houver alarmes ativos em um grupo. Os atributos associados são:

- Nenhum = 0: sem som
- Bip = 1: um bip regular será tocado em cada computador cliente enquanto houver alarmes sem reconhecimento

Show

Configura a exibição ou não do alarme (inclui a visualização no objeto alarmes online).

Log Events

Define o tipo de arquivamento de Historiador em eventos de Alarme. Os seguintes tipos são possíveis:

- None (sem log)
- Active (log quando o evento estiver ativo)
- ActiveAck (log quando o evento estiver ativo e for reconhecido)
- ActiveNorm (log quando o evento retorna ao normal)
- All (log em todas as condições acima)

Nota:

O banco de dados na qual os tags são salvos é definido em EditDatasetDBs no DB denominado "AlarmHistorian".

Colors

Define a customização da tela de Alarme para cada linha de Alarme de acordo com seu grupo principal.

Itens de Alarme

Configura tags para gerar alarmes sob condições definidas.

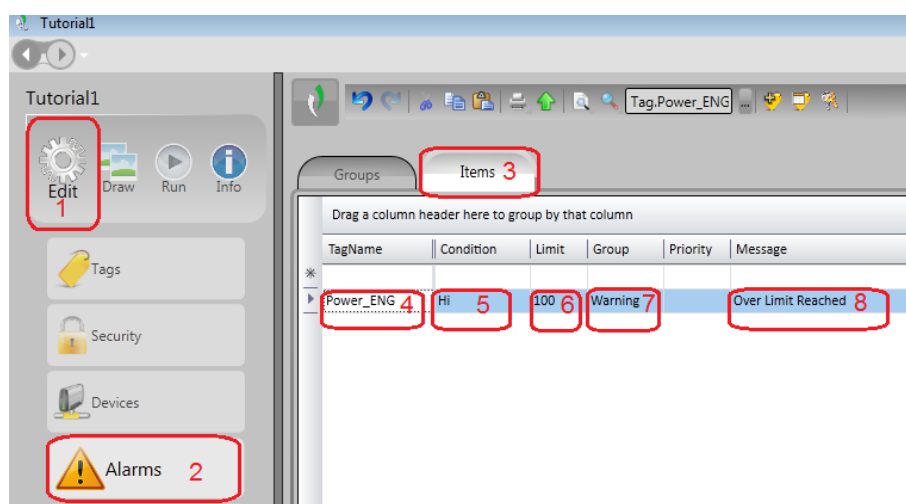



Figura 6-18. Itens de Alarmes

Os passos para editar ou incluir os grupos de alarmes estão ilustrados na Figura 6 18. e descritos a seguir:

1. Selecionar o menu Edit
2. Acessar o item Alarms
3. Clicar na guia Itens para acesso as configurações dos itens de alarmes
4. Incluir ou editar o nome do tag que gera o alarme na linha marcada com um asterisco (*)
5. Selecionar a condição de geração de alarme
6. Configurar o limite que dispara o alarme
7. Selecionar o grupo previamente criado com as características do alarme
8. Inserir mensagem que será apresentada caso o alarme seja disparado

Name

Define o TagName que será avaliado para gerar o Alarme.

Condition

Condição de avaliação para gerar alarmes. Opções...

- Hi: Tag >= limite
- HiHi: Tag >= limite (quando reconhecido automaticamente, reconhece o alarme Hi para o mesmo Tag)
- Lo: Tag <= limite
- LoLo: Tag <= limite (quando reconhecido automaticamente, reconhece o alarme Lo para o mesmo Tag)
- RateOfChange: Taxa de alteração do Tag >= limite
- DeviationMinor: Valor absoluto (Tag - Setpoint) > limite (Setpoint definido na coluna Setpoint)
- DeviationMajor: Valor absoluto (Tag - Setpoint) > limite (Setpoint definido na coluna Setpoint)
- Equal: Tag = limite
- GreaterThan: Tag > limite
- GreaterEqual: Tag >= limite
- LessThan: Tag < limite
- LessEqual: Tag <= limite
- Changed: valor do Tag foi alterado
- ChangedUp: valor do Tag aumentou
- ChangedDown: valor do Tag diminuiu

Limit

Define o valor para avaliar as condições do alarme.

Group

Define o nome do Grupo no EditAlarmGroups onde o comportamento do item Alarme está especificado. Os grupos pré-definidos são:

- Critical (mensagens críticas que exigem reconhecimento)
- SystemEvent (log de eventos de sistema)
- Warning (mensagens de advertência que não exigem reconhecimento)

Priority

Define a Prioridade do Alarme. "0" é a prioridade mais alta.

Message

Define uma Mensagem a ser exibida quando ocorrer o Alarme. Campos de mensagem podem conter valores de Tag avaliados quando da geração do Alarme. Para tanto acrescente o nome do Tag desejado na mensagem usando a notação de vinculação XAML entre chaves, conforme ilustrado no exemplo a seguir:

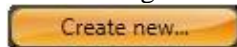
```
ValordoTag = {NomedoTag.Value}.
```

Editando Banco de Dados

Conforme mencionado anteriormente, o módulo de conjunto de dados (Datasets) incluído no BluePlant fornece uma interface de fácil operação para a troca de dados em tempo real com bancos de dados externos, XML, CSV ou arquivos de texto, bem como a possibilidade de acessar tabelas e

consultas SQL. O banco de dados em tempo real garante, sem a necessidade de qualquer programação adicional, a sincronização de dados entre vários processos no servidor e múltiplas estações cliente. Um vasto conjunto de propriedades internas, tais como qualidade de dados, estampa de tempo, bloqueio de estado e valor bloqueado simplificam a criação de aplicações.

DBs

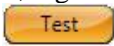
Através da guia DBs é possível acessar a configuração do banco de dados. A partir do botão Create new... é possível selecionar o provedor do banco de dados e uma nova conexão.

Name

Estabelece o nome para o DB.

Provider

Seleciona o fornecedor para o DB quando da criação de uma nova conexão de base de dados. Os fornecedores padrão são:

- Odbc Data Provider - Permite o acesso a um banco de dados ODBC através de um driver ODBC nativo
- OleDb DataProvider - Fornece aplicativos para acesso aos dados armazenados em diversas fontes de informação
- SqlClient Data Provider - É uma coleção de classes que podem ser usadas para acessar bancos de dados SQL Server
- Microsoft SQL Server Compact Data Provider - Fornece acesso aos bancos de dados Microsoft SQL Server Compact
- TatsoftDB 4 direct connection – Permite acesso ao banco de dados TatsoftDB 4

Outros fornecedores podem ser criados adicionando-se modelos XML ao produto na subpasta "DBProviders" localizada na pasta de instalação do software BluePlant. Exemplos de provedores de dados:

- OdbcDataProvider
- Firebird database
- Microsoft Access Database
- Microsoft Excel Database
- ODBC using DSN
- ODBC using fileDSN
- Oracle Database
- SQL Server Database

Database

Seleciona o banco de dados quando da criação do objeto DB. A lista de base de dados disponíveis é criada dinamicamente baseada em fornecedores encontrados na subpasta "DBProviders" localizada na pasta de instalação do software BluePlant. As bases de dados mais comuns são:

- Microsoft Access Database
- Microsoft Excel Database
- ODBC DSN
- ODBC FILEDSN
- SQL Server Database

Connectionstring

Esta coluna permite definir a string de conexão usada para comunicar-se com o banco de dados. Para tanto, digite o caminho da fonte de dados onde está localizado o arquivo do banco de dados. Clique em Test para testar se a fonte de dados foi localizada e está válida.

NameLogon

Define o nome do logon para conectar-se com o banco de dados.

Password

Senha solicitada para conectar-se com o banco de dados. Este campo pode ser editado somente pelo usuário Administrador (usuário ID: 2).

Conexão com Excel

Conecta o Excel usando um driver ODBC, ODBC DSN ou OleDB. Para a conexão via ODBC deve-se proceder conforme segue:

1. Selecione e nomeie uma faixa de linhas e colunas na planilha. Isto permitirá ao BluePlant ler as informações como uma tabela
2. Escolha um dos seguintes processos de nomeação de acordo com a versão do Microsoft Excel

No Microsoft Office 2007: clique com o botão direito na seleção e escolha "Nomear uma Faixa". A Figura 6-19 ilustra essa opção.

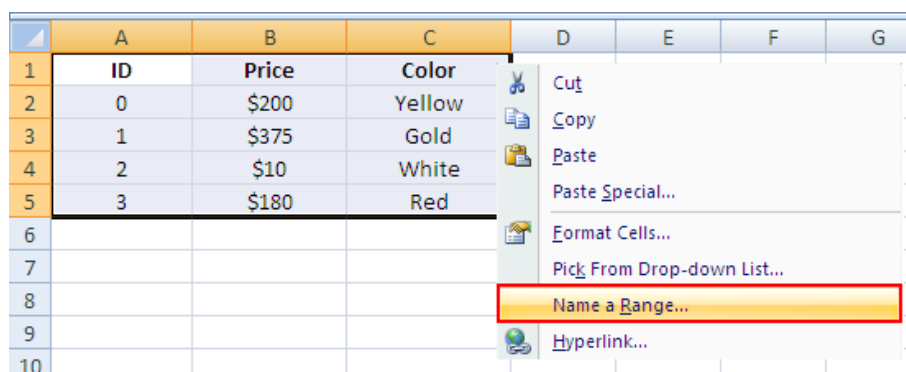



Figura 6-19. Conexão com Excel 2007

No Microsoft Office 2003: no Microsoft Excel vá para "Inserir > Nome > Definir…". A Figura 6-20 ilustra essa opção.

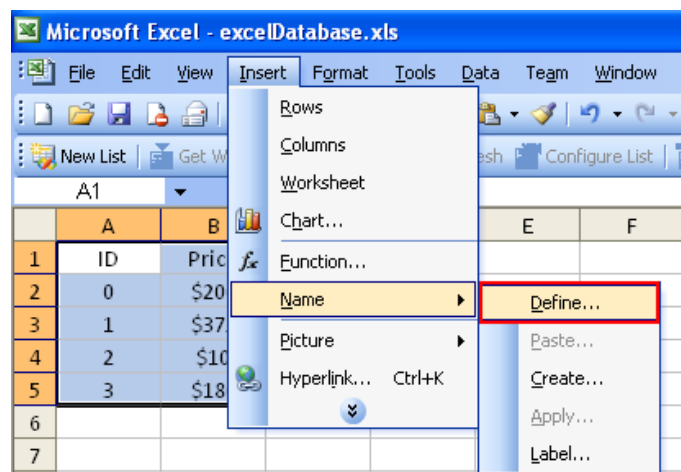



Figura 6-20. Conexão com Excel 2003

Dê um nome à seleção (por exemplo, "itemsTable"). O arquivo Excel está pronto para uso.

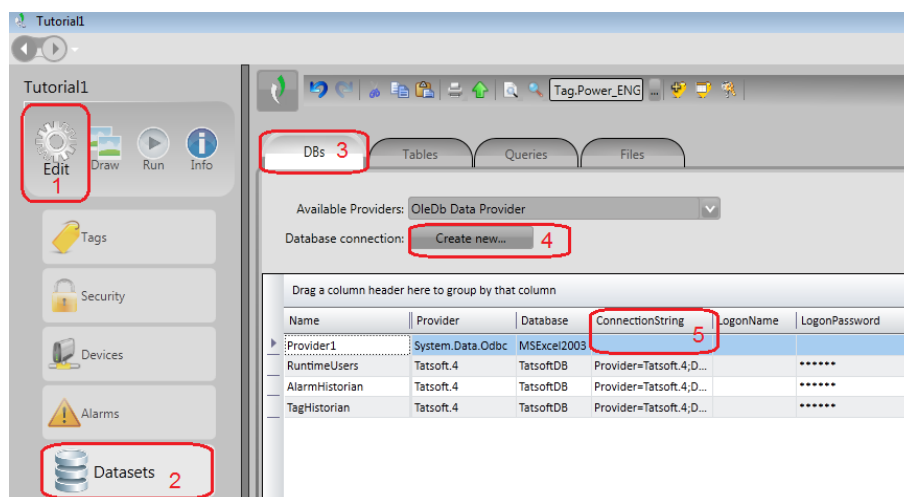



Figura 6-21. Configuração da Conexão com Provider Excel 2003

Os passos para criar uma nova conexão com banco de dados estão ilustrados na Figura 6-21 e descritos a seguir:

1. Selecionar o menu Edit
2. Acessar o item Datasets
3. Clicar na guia DBs para acesso as conexões criadas com bancos de dados
4. Botão para criar uma nova conexão com Excel 2003, conforme pode ser visto na Figura 6-22
5. Na coluna Connectionstring é onde se configura a conexão e teste com o banco de dados. Na Figura 6-23 pode ser visualizada a configuração e teste

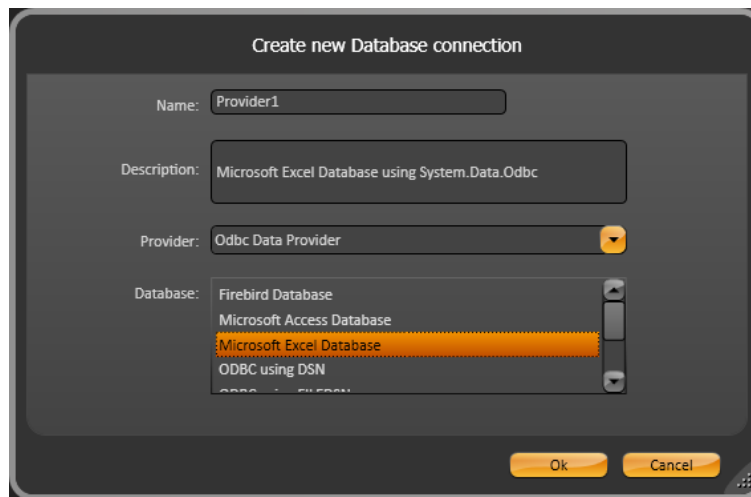



Figura 6-22. Criar uma Nova Conexão com Banco de Dados

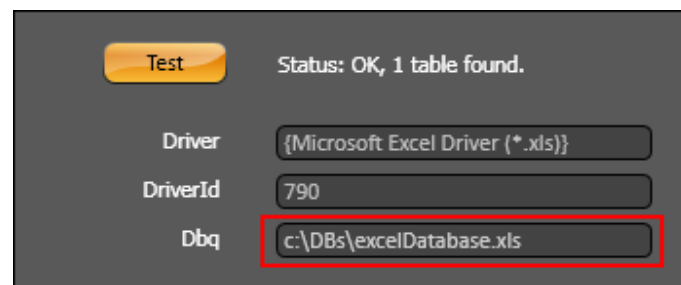



Figura 6-23. Teste da Conexão com Banco de Dados

Por outro lado, para configurar uma base de dados do tipo ODBC DSN proceda conforme indicado a seguir:

No painel de controle do computador selecione "Administrative Tools" e dê um clique duplo em "Data Sources (ODBC)", conforme Figura 6-24. No caso onde o sistema operacional for Windows 64 bits, o acesso é através do arquivo C:\Windows\SysWOW64\odbccad32.exe.

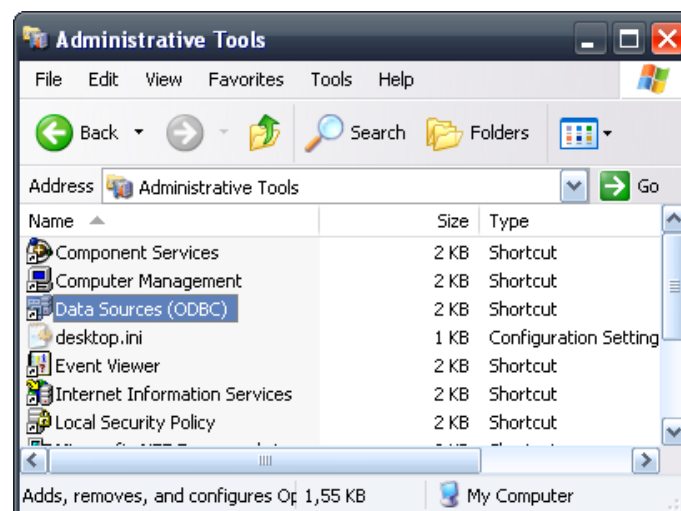



Figura 6-24. ODBC com DSN

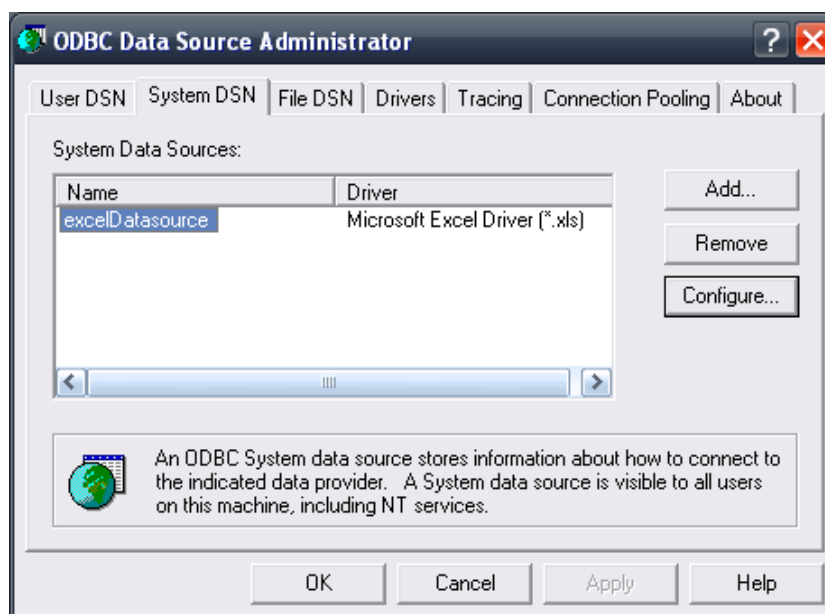



Figura 6-25. "ODBC Data Source Administrator"

Na janela "ODBC Data Source Administrator" clique na opção de “Add...”. Será solicitado que seja selecionado um driver. Selecione o "Microsoft Excel Driver (*.xls)" e seguinte tela deve aparecer conforme a Figura 6-26.

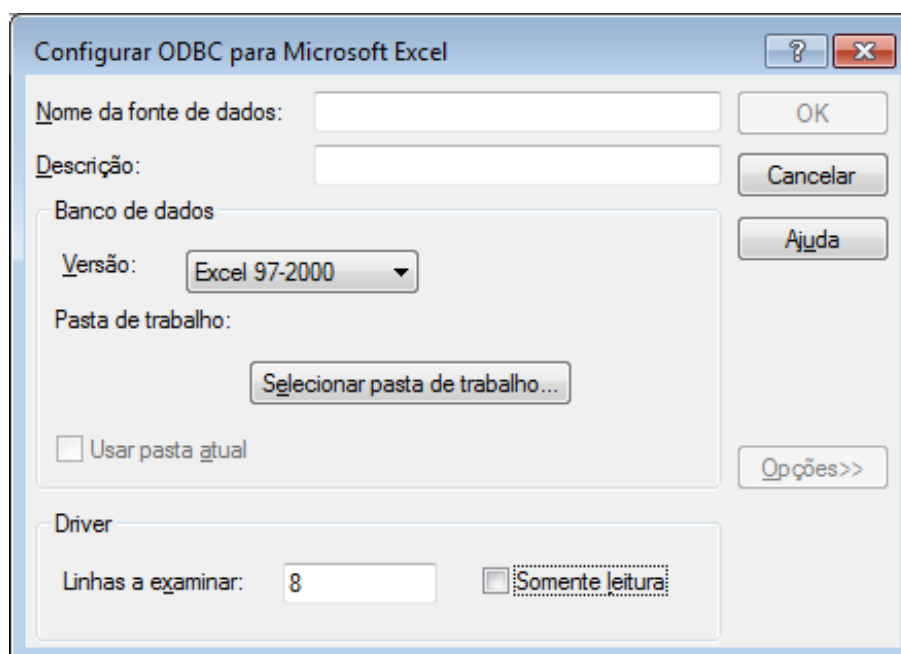



Figura 6-26. Configuração da base de dados

Clique em “Selecionar pasta de trabalho” e selecione o nome do arquivo Excel criado anteriormente. (por exemplo, "excelDatasource"). Para acesso de Escrita desmarque a checkbox "Somente leitura".

No namespace do Datasets do BluePlant, escolher a guia "DBs" e criar um novo provider na opção correspondente. Nas opções do "Odbc Data Provider" escolha "ODBC using DSN" e clique em Ok. Na coluna Connectionstring da nova linha, entre com DSN no campo apropriado e clique no botão teste para verificar a conexão com o Excel.

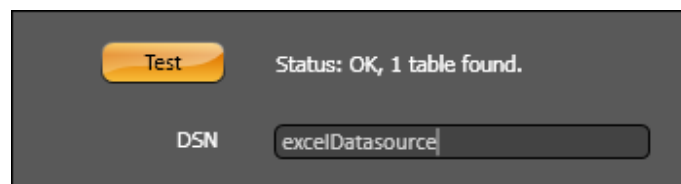



Figura 6-27. Teste da Conexão à Base de Dados ODBC

Para conectar o provider OLEDB com o Excel, pode ser feito através dos passos abaixo:

No namespace do Datasets do BluePlant, escolha a guia "DBs". Selecione a opção "OleDb data provider" na caixa-combo e crie um novo provider na opção correspondente.

Escolha "Microsoft Excel Database" e clique em Ok. Clique na coluna Connectionstring do novo provider inserido, entre com o caminho e o nome do arquivo Excel (.xls) no campo "DataSource" conforme mostrado na Figura 6-28.

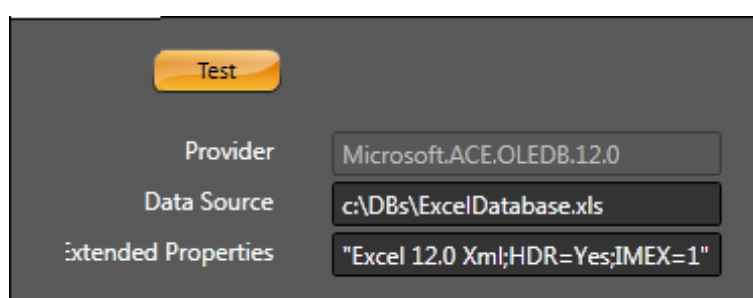



Figura 6-28. Teste da Conexão à Base de Dados OLEDB

Conexão com Banco de Dados Oracle

O provider Oracle Oracle.DataAccess.Client utilizado no exemplo de conexão com banco de dados Oracle, está disponível para download gratuito no seguinte endereço:

http://www.oracle.com/technology/software/tech/windows/odpnet/index.html

Para criar uma conexão com banco de dados Oracle clique em Create new... e selecione o provider instalado conforme mostrado na Figura 6-29.

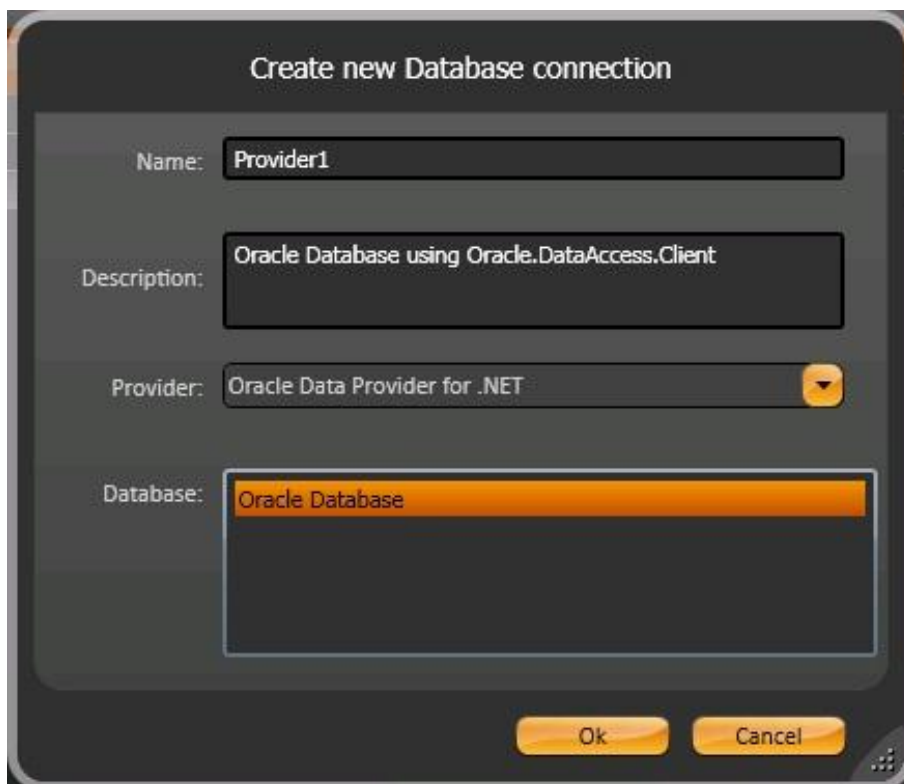



Figura 6-29. Inserindo Provider Oracle

Deve ser inserido o usuário e senha do banco de dados para garantir que a conexão seja estabelecida. Para a inserção as colunas LogonName e Logon Password devem ser preenchidas com usuário e senhas previamente cadastradas no banco de dados.

Figura 6-30. Usuário e Senha para Conexão com Banco de Dados Oracle

Acessar os campos da coluna ConnectionString para inserir as configurações do endereço IP, porta e o SID do banco para estabelecer a conexão, conforme pode ser visto na Figura 6-31.

Figura 6-31. Configuração do Endereço IP, Porta e SID do Banco de Dados Oracle

Depois de configurado o endereço IP, porta e SID, para verificar se a conexão com o banco de dados Oracle está funcionado, basta clicar no botão “Test” e deve aparecer o resultado semelhante ao da Figura 6-32.

Figura 6-32. Teste de Conexão com Banco de Dados Oracle

ATENÇÃO: O provider utilizado no exemplo para conexão com banco de dados Oracle, requer clientes Oracle na versão 9.2 ou anteriores. O provider Oracle que acompanha o BluePlant é o System.Data.OracleClient, requer clientes Oracle na versão 8.1.7 ou superiores.

Conexão com Banco de Dados SQLServer

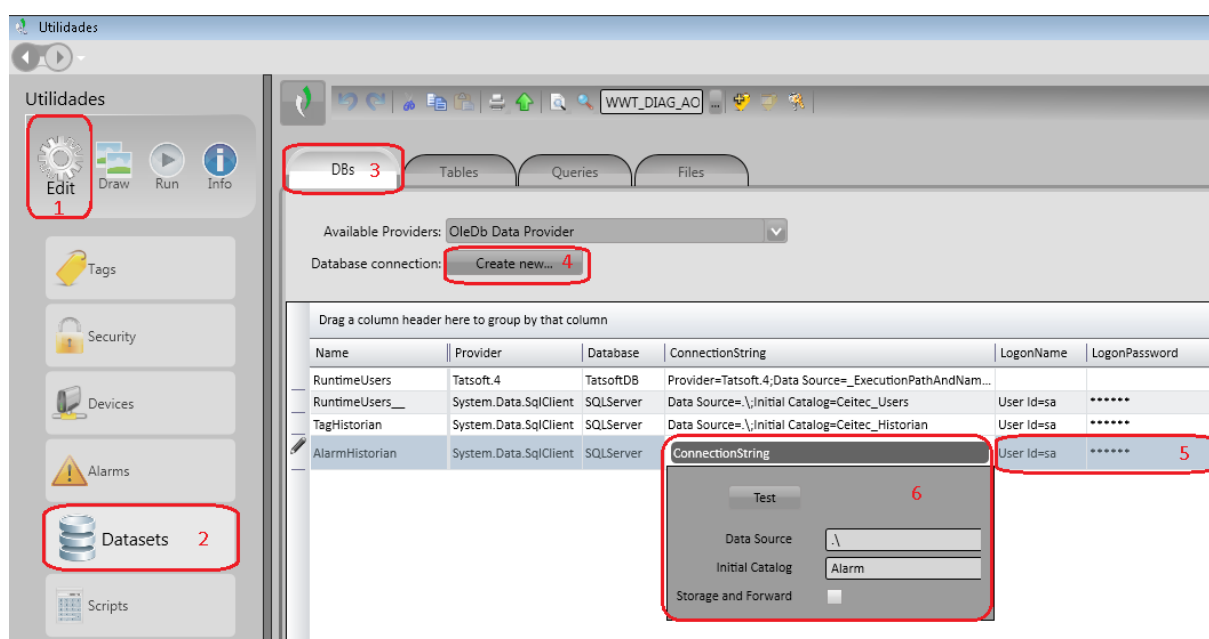



Figura 6-33. Inserção de um Provider SQLServer

Os passos para criar uma nova conexão com banco de dados SQLServer estão ilustrados na Figura 6-33 e descritos a seguir:

1. Selecionar o menu Edit
2. Acessar o item Datasets
3. Clicar na guia DBs para acesso as conexões criadas com bancos de dados
4. Através do botão Crate new..., criar uma nova conexão com banco de dados SQLServer
5. Configurar o usuário e senha do banco de dados nas colunas LogonName e LogonPassword
6. Na coluna Connectionstring deve ser configurada a conexão e realizar teste de conexão com o banco de dados SQLServer

Tables

A guia Tables permite acessar as tabelas de dados a partir das bases de dados configuradas (DBs). A Figura 6 27 ilustra essa seleção.

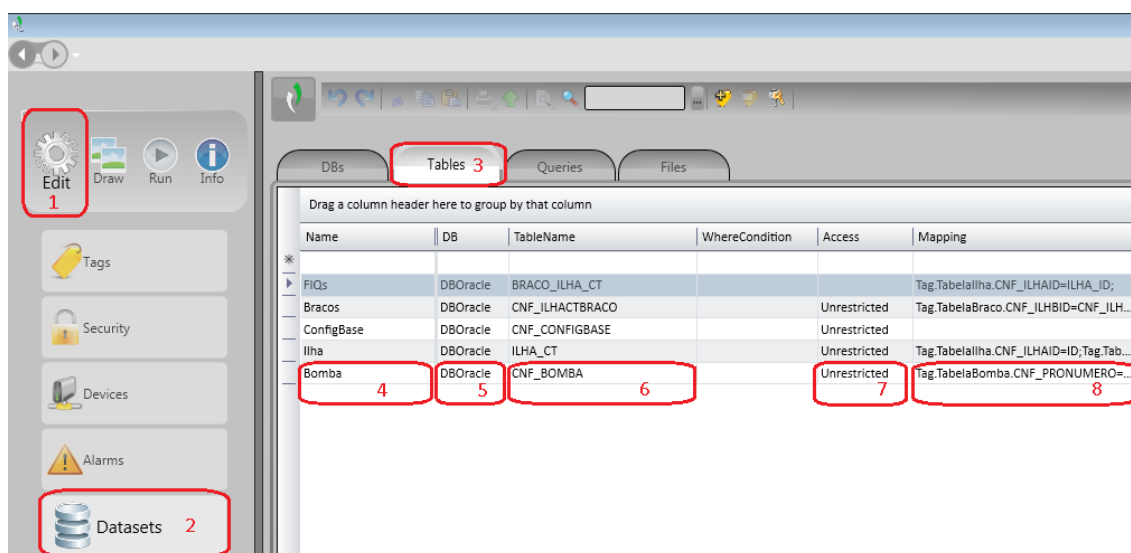



Figura 6-34. Edição de Tabelas de Dados

Os passos para editar ou incluir tabelas estão ilustrado na Figura 6-34 e descritos a seguir:

1. Selecionar o menu Edit
2. Acessar o item Datasets
3. Clicar na guia Tables para acesso às configurações das tabelas
4. Incluir ou editar o nome da tabela na linha marcada com um asterisco (*)
5. Selecionar qual conexão com o banco de dados previamente criado será utilizada
6. Selecionar qual a tabela do banco de dados que irá comunicar
7. Selecionar qual a permissão de acesso ao banco de dados para a tabela que configurada
8. Configurar na coluna Mapping os tags relacionados com as colunas da tabela do bando de dados

Name

Indica o nome da Tabela usada nos objetos Runtime.

DB

Indica o nome da DB (conexão da Base de Dados) a qual pertence à tabela de dados.

TableName

Indica o nome da tabela assim como no banco de dados.

WhereCondition

Indica qual a condição em que a instrução SQL será executada.

Access

Define a política de segurança com operações permitidas para esta DataTable. As opções são:

- Read (leitura)
- Insert (inserção)
- ReadWrite (leitura e escrita)
- Unrestricted (irrestrita)

Mapping

Mapeia os conteúdos da tabela de dados com os objetos Tag. Ao executar os comandos de seleção e próximo, a primeira linha no resultado selecionado é aplicada aos tags. Então, ao executar o comando de atualização, os conteúdos do Tag são escritos na linha.

Queries

Esta guia mapeia as bases de dados configuradas (DBs) conforme mostrado na Figura 6-35.

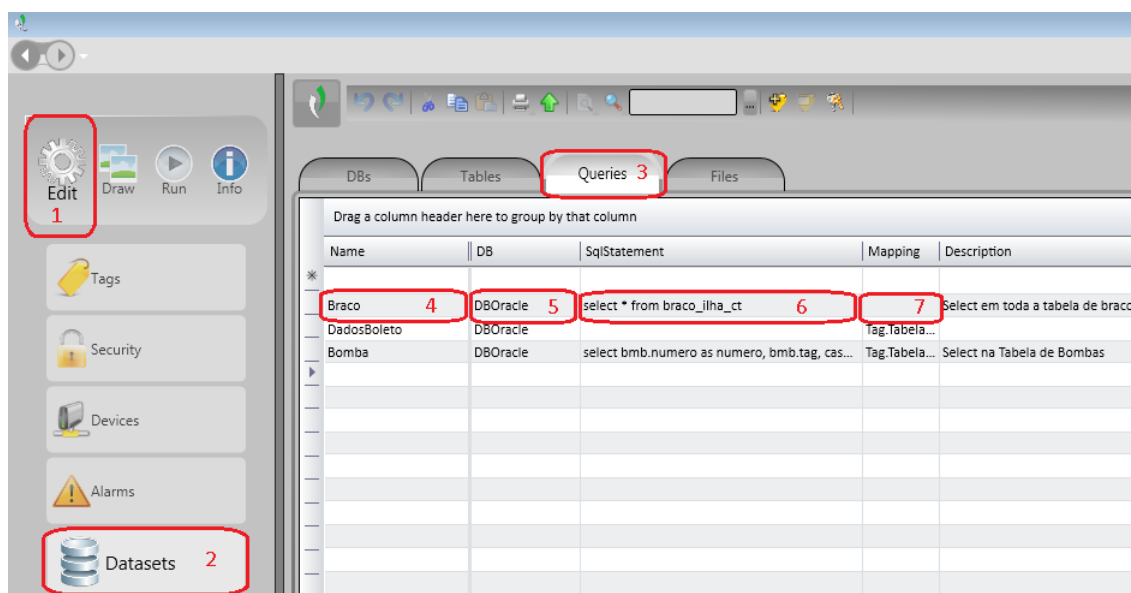



Figura 6-35. Edição de Queries

Os passos para editar ou incluir instruções SQL(queries) estão ilustrados na Figura 6-35 e descritos a seguir:

1. Selecionar o menu Edit
2. Acessar o item Datasets
3. Clicar na guia Queries para acesso às configurações das instruções SQL
4. Incluir ou editar o nome da instrução na linha marcada com um asterisco (*)
5. Selecionar qual conexão com o banco de dados previamente criado que será utilizada
6. Configurar a instrução SQL na coluna SqlStatement
7. Configurar na coluna Mapping os tags relacionados às colunas da tabela do bando de dados.

Name

Nome para o objeto Query usado nos objetos de Runtime.

DB

Nome da conexão do banco de dados usada para o armazenamento dos queries.

SQLStatement

Instrução SQL utilizada para realizar consultas no banco de dados.

Mapping

Mapeia os conteúdos da tabela de dados em tags. Ao executar o comando de seleção, a primeira linha no resultado é aplicada aos tags.

Files

A guia Files define o(s) arquivo(s) para a troca de informações com os bancos de dados conforme ilustrado na Figura 6-36.

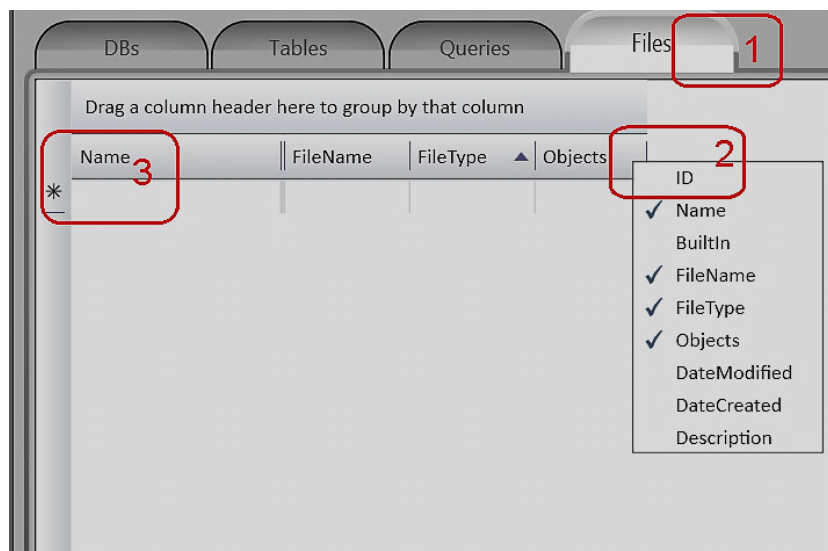



Figura 6-36. Edição Arquivos de Bancos de Dados

Os itens que compõem o menu edição de arquivos de bancos de dados estão numerados na figura anterior e são descritos na sequência:

1. Selecionar a opção files clicar com o botão esquerdo do mouse
2. Clicar com o botão direito do mouse em um cabeçalho de coluna para visualizar os itens desejados
3. Observar que os dados relativos aos arquivos irão aparecer nas linhas da tabela

Na sequência são explicados os itens de edição de arquivos de bancos de dados conforme a ilustração da Figura 6-36.

Name

Define o nome do arquivo usado nos objetos de Runtime.

FileName

Nome do arquivo e caminho. Exemplo: C:\BDs\file1.txt.

FileType

Caracteriza os tipos de arquivo. Os tipos de arquivo possíveis são:

- ASCII (codificação de caracteres de sete bits baseada no alfabeto inglês)
- Unicode (representação e manipulação de texto)
- XML (linguagem de marcação para a criação de documentos com dados organizados hierarquicamente)

Objects

Mapeia os conteúdos do arquivo com objetos Tag.

Editando Scripts

Um script pode ser executado na ocorrência de um evento de disparo ou ciclicamente através da definição de um intervalo de tempo para tal. As opções que compõem o menu de edição de scripts estão detalhadas na sequência.

Tasks

A Figura 6-37 enfatiza a aba Tasks para edição de tarefas de script. Existem quatro tarefas incorporadas:

- ServerStartup – o script é executado na máquina do Servidor (TServer.exe) quando o projeto entra em execução
- ServerShutdown - o script é executado na máquina do Servidor na parada da operação
- ClientStartup - o script é executado em cada máquina cliente quando o TVisualizer.exe (módulo de telas) inicia sua execução
- ClientShutdown - o script é executado em cada máquina cliente quando o módulo de telas é encerrado

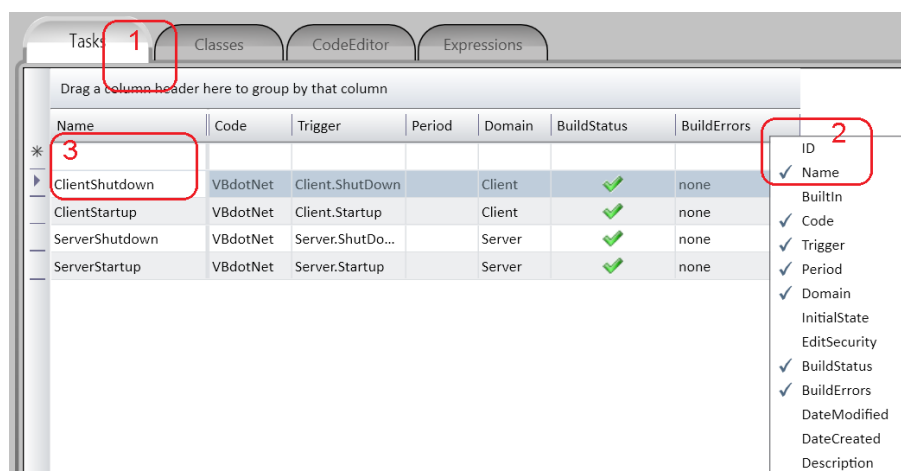



Figura 6-37. Edição de Tarefas de Script

Os itens que compõem o menu de edição de tarefas de script estão numerados na figura anterior e são descritos na sequência:

1. Selecionar a guia Tasks
2. Selecionar as opções de configuração das colunas da tabela desejadas clicando com o botão direito do mouse no cabeçalho de uma coluna e marcando os itens para exibição
3. Observar: os dados relativos às tarefas de script irão aparecer nas linhas da tabela

Na sequência são explicados os itens do menu edição de tarefas de script conforme a ilustração da Figura 6-37.

Name

Indica o nome para o objeto Script.

Code

Define a linguagem do Script. O usuário pode selecionar entre VBdotNet ou CSharp.

Trigger

Tag ou objeto que dispara a execução da Tarefa. A tarefa é executada quando o valor do objeto muda.

Period

Período de tempo para executar uma Tarefa.

Domain

Define se o Script é executado na estação do Servidor ou em cada estação Cliente.

BuildStatus

Indica o status da última compilação: verde significa compilado com sucesso e vermelho indica erro de compilação

BuildErrors

Status de contagem de erro a partir da última compilação do Script. Atributos associados: ReadOnly (somente leitura).

Classes

Nesta opção estão listadas as classes definidas pelo usuário via biblioteca de métodos. A Figura 6-38 mostra esse item do menu Script. Há duas UserClasses incorporadas:

- ServerMain: biblioteca de métodos disponível para todas as tarefas do servidor e clientes
- ClientMain: biblioteca de métodos disponível para todos clientes scripts, inclusive telas scripts

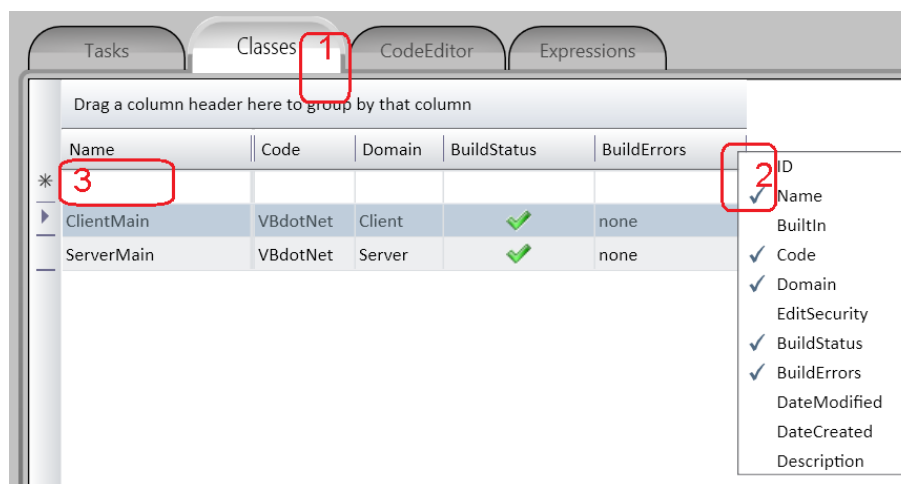



Figura 6-38. Classes de Script

Os itens que compõem o menu de edição de classes de script estão numerados na figura anterior e são descritos na sequência:

1. Selecionar a guia Classes
2. Selecionar as opções de configuração das colunas da tabela desejadas clicando com o botão direito do mouse no cabeçalho de uma coluna e marcando os itens para exibição
3. Observar: os dados relativos às classes de script irão aparecer nas linhas da tabela

Na sequência são explicados os itens do menu de edição de tarefas de script conforme a ilustração da Figura 6-38.

Name

Indica o nome da classe do usuário.

Code

Indica a linguagem do Script. O usuário pode selecionar entre VBdotNet ou CSharp.

Domain

Define se os métodos nessa classe de usuário serão visíveis para os scripts do Servidor ou Cliente.

CodeEditor

Trata-se do painel editor de código das tarefas e classes de usuário conforme ilustrado na Figura 6-39 (Item 1). As opções de linguagem de programação são: VBdotNet ou CSharp. A seleção da linguagem está ilustrada na Figura 6-39 (Item 2). Informações sobre a sintaxe das linguagens de programação podem ser obtidas no seguinte endereço http://www.microsoft.com/net.

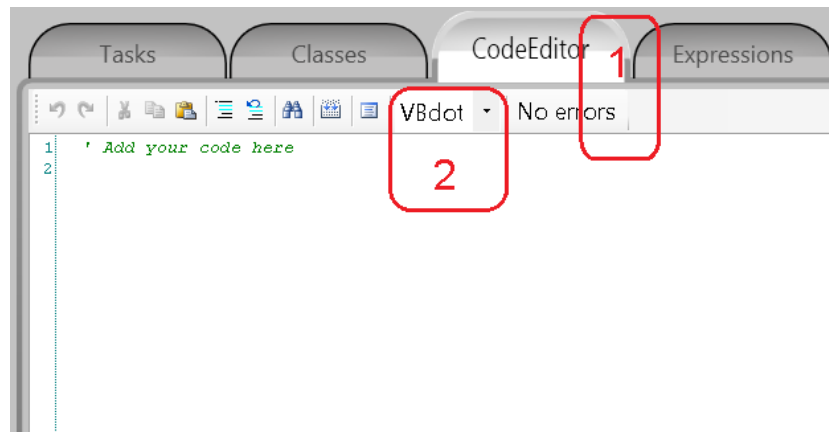



Figura 6-39. CodeEditor

Utilizando Tarefas de Script

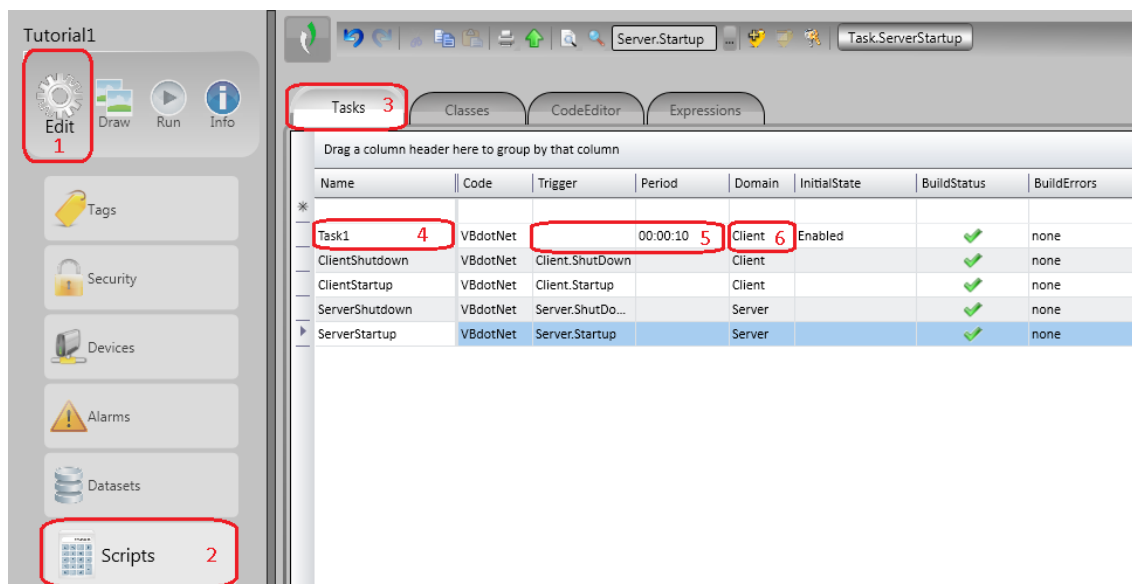



Figura 6-40. Tarefas de Scripts

Os passos para editar ou incluir uma tarefa de script estão ilustrados na Figura 6-40 e descritos a seguir:

1. Selecionar o menu Edit
2. Acessar o item Scripts
3. Clicar na guia Tasks para acesso às tarefas de scripts configuradas no sistema
4. Incluir ou editar o nome da tarefa na linha marcada com um asterisco (*)
5. Configurar a variável de disparo ou o período que a tarefa de script será executada, através das colunas Trigger ou Period
6. Selecionar em qual ambiente (cliente ou servidor) a tarefa será executada

A Figura 6-41 a seguir apresenta os passos para a codificação da tarefa de script chamando o método de uma classe existente.

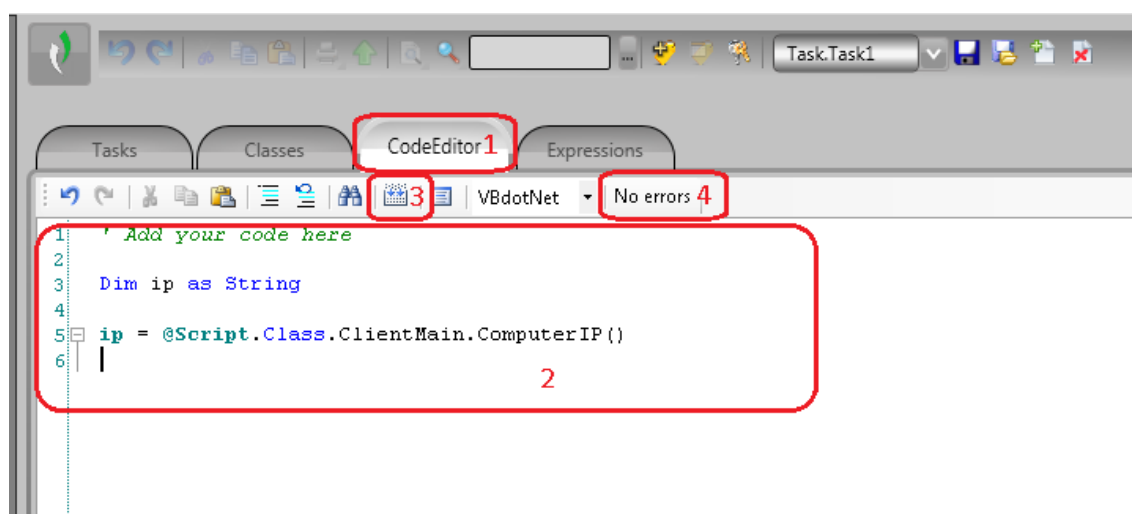



Figura 6-41. CodeEditor das Tarefas de Scripts

1. Clicar na guia CodeEditor para acesso ao ambiente de codificação
2. Digitar o código no espaço demarcado
3. Clicar no ícone para compilar o código
4. Verificar a existência de erros no script codificado

ATENÇÃO: Para utilizar os namespaces no ambiente de codificação é obrigatório digitar o caractere @ antes do namespace, caso contrário será utilizado como uma variável local gerando possíveis erros na compilação.

Editando Displays

A opção Displays do menu Edit permite a configuração do módulo associado às telas. Esse módulo contempla o repositório das telas (Displays), seus modos de operação (Layouts) e imagens (Resources).

Displays

A guia Displays, quando selecionada, lista as telas que compõem o projeto. A visualização da lista de telas pode ser no formato de tabela ou cartão. O seletor Card View/Table View permite alternar o modo de visualização. A Figura 6-42 ilustra esse ambiente no formato de tabela.

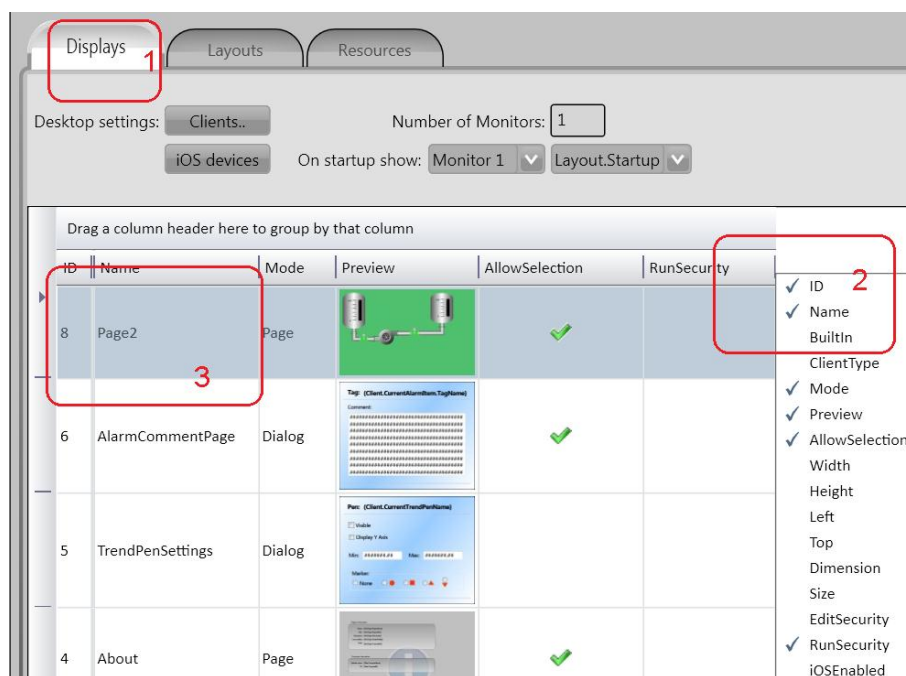



Figura 6-42. Edição de Displays

Os itens que compõem o menu de edição de telas estão numerados na figura anterior e são descritos na sequência:

1. Acesso à guia Displays
2. Clicar com o botão direito do mouse em qualquer um dos cabeçalhos de coluna para selecionar as opções de configuração a serem exibidas
3. Clicar na linha da tela para alterar as configurações

Name

Nome da tela.

Mode

Modo de operação da tela (Página, Diálogo ou Popup). Essas opções são detalhadas na sequência:

- Página: este é o modo de exibição padrão. Quando uma Página é aberta automaticamente, a última página do layout atual é fechada
- Diálogo: uma tela de Diálogo se abre como um "Window Modal Dialog". Isto significa que os comandos em todas as outras telas abertas ficam desabilitados. Ao clicar em OK no diálogo o método On OK na tela CodeBehind é executado
- Popup: uma tela de popup abre-se acima das outras telas. Ao abrir uma nova Página, por padrão, todas as telas de popup são fechadas

Preview

Visualização da imagem da tela.

AlowSelection

Marca indicando se a tela está listada no objeto DisplaySelection quando em execução. AllowSelection é uma função utilizada pelo criador da aplicação/projeto para remover a seleção de página, teste e outras telas do operador padrão e assegurar que os itens removidos estejam disponíveis somente para contextos específicos da aplicação. O método incorporado para definir a seleção da tela ao rodar a aplicação é a função "PageSelecter" que é configurada na guia Editar>Telas>Telas. Para impedir essa seleção pelo operador padrão remova o marcador da página ou da opção "AllowSelection" na coluna dos objetos.

RunSecurity

Grupos de permissões de segurança exigidos para abrir esta tela durante o Runtime. Clique no campo "Run Security" para abrir a janela "Run Security Selection". Após concluir as seleções clique em qualquer lugar na tabela para fechar a janela de seleção. Suas seleções aparecerão no campo "RunSecurity" da linha da tabela.

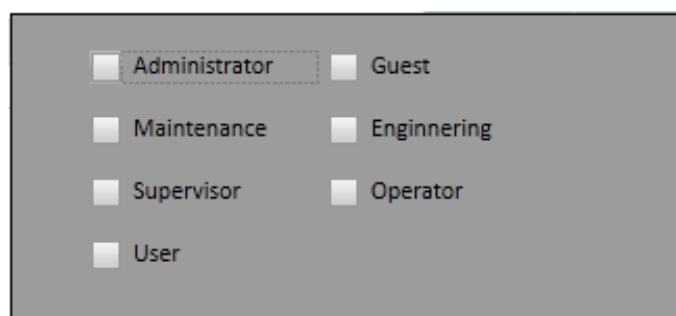



Figura 6-43. Permissões no Runtime

Configurando Múltiplos Monitores

Os passos para configurar múltiplos monitores estão ilustrados na Figura 6-44 e descritos abaixo:

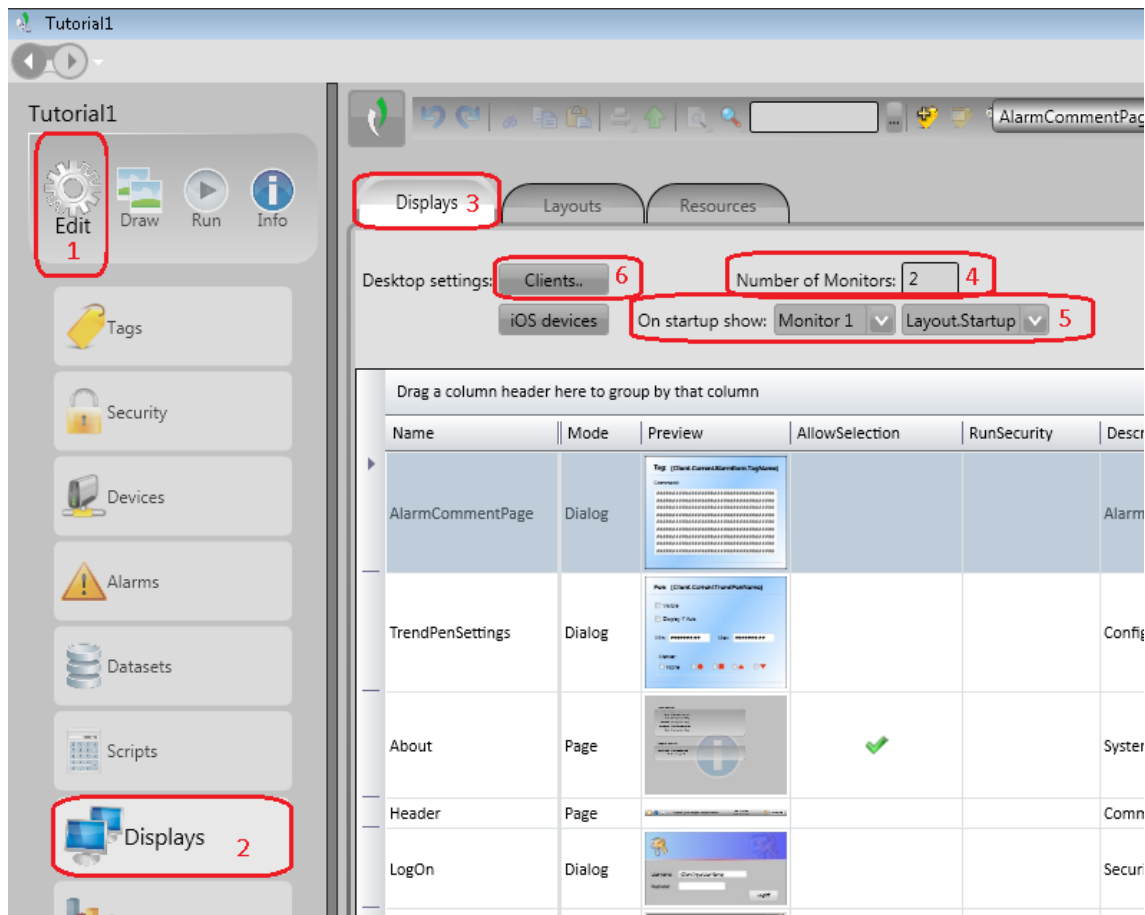



Figura 6-44. Configurando Múltiplos Monitores

1. Selecionar o menu Edit
2. Acessar o item Displays
3. Clicar na guia Displays para acesso as configurações das telas já existentes no projeto
4. Configurar a quantidade de monitores que deseja utilizar
5. Selecionar para cada monitor o layout correspondente que irá abrir quando executar o projeto
6. Clicar no botão Clients a realizar a configuração dos Clientes rich no Runtime (uma configuração única para todos os clientes rich). Conforme Figura 6-45

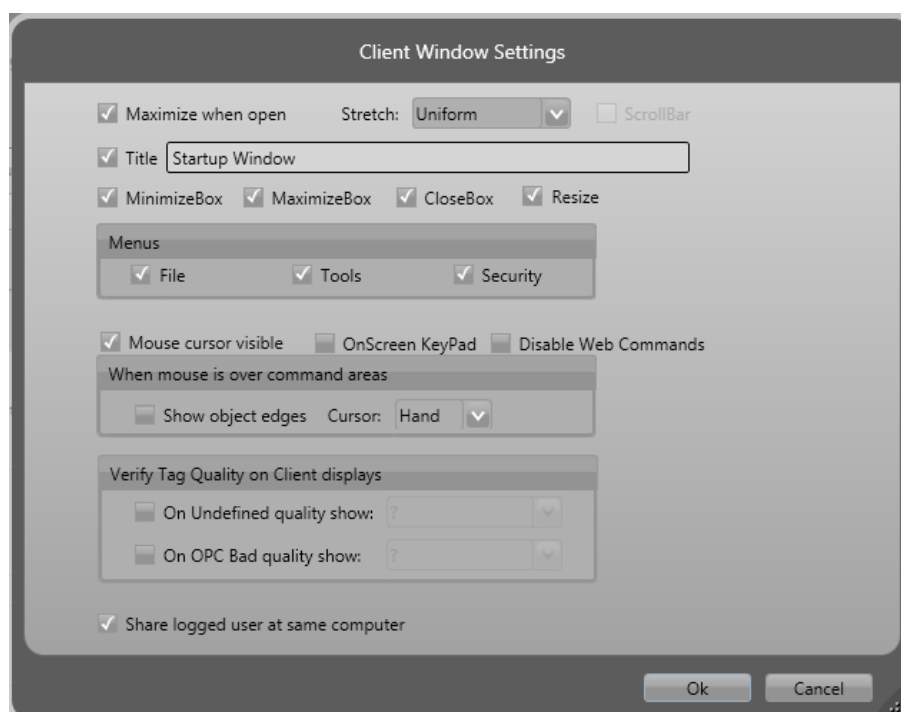



Figura 6-45. Configurações dos Clientes Rich

Layouts

Edita os layouts do projeto. Layouts são containers (DockPanels) que definem o modo da tela durante a execução. A última tela listada no layout é aquela que será alterada sob o comando Client.OpenDisplay(). As demais que permanecem abertas normalmente contêm informações tais como menus de navegação, linhas de alarme e informações de aplicação global. O projeto pode ter apenas um layout, ou pode ser alterado dinamicamente através do comando Client.OpenLayout().

Clique New para criar um novo layout. Clique + ou − para acrescentar ou excluir páginas no layout selecionado e ↑↓ para navegar para cima e para baixo na lista da página. As dimensões do layout podem ser definidas usando os campos largura e altura, podendo alterar o fundo para uma cor ou imagem.

A Figura 6-46 ilustra a configuração dos layouts.

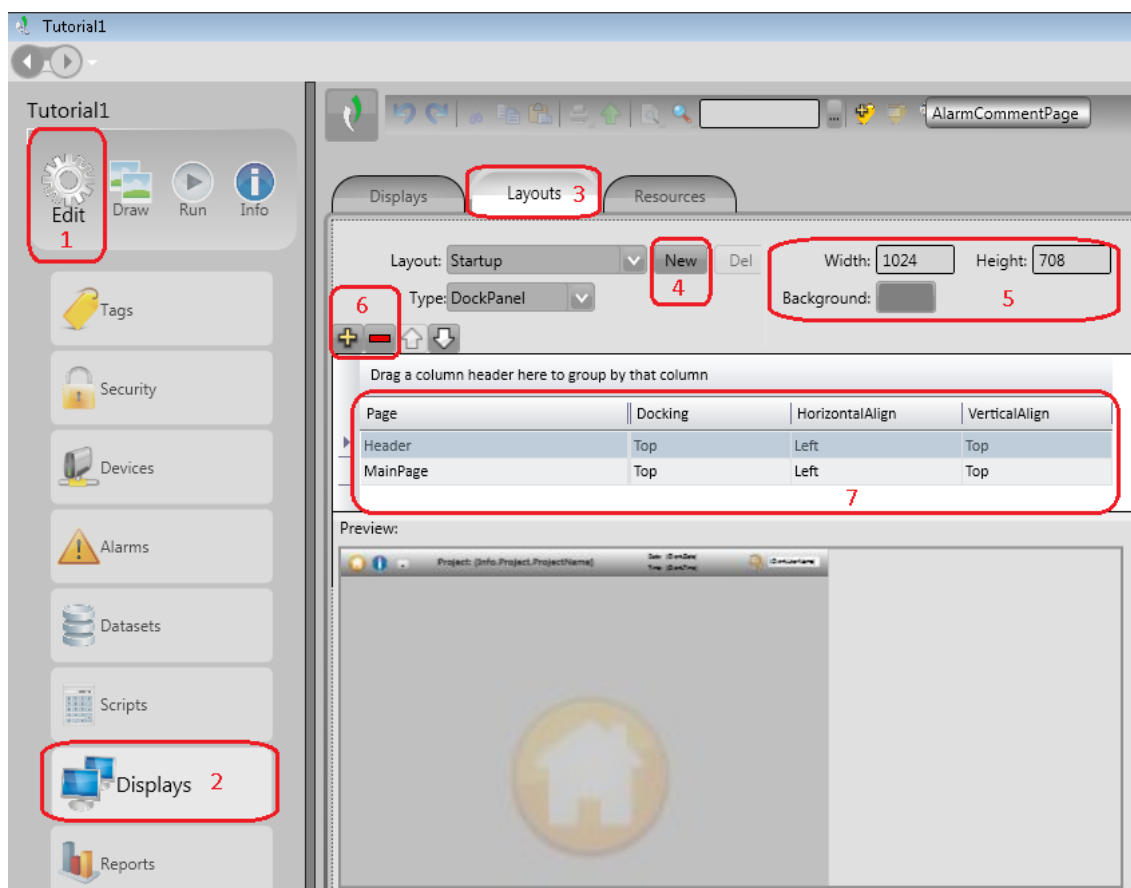



Figura 6-46. Edição de Layouts

Os passos para editar e incluir um novo layout estão ilustrados na Figura 6 46 descritos a seguir:

1. Selecionar o menu Edit
2. Acessar o item Displays
3. Clicar na guia Layouts para acesso as configurações dos layouts já existentes no sistema.
4. Clicar no botão New para adicionar um novo layout
5. Configurar as dimensões e cor de fundo do layout correspondente
6. Configurar a quantidade de telas que o layout irá apresentar utilizando os botões .
7. Selecionar as telas e suas posições no layout

Page

Nome da tela usada neste Layout. Somente telas do modo PÁGINA podem ser incluídas em layouts.

Docking

Define a posição de encaixe da Tela. As opções possíveis são:

- Esquerda
- Topo
- Direita
- Inferior

A posição de encaixe pode ser alterada clicando-se uma vez no campo da tabela e selecionando-se a posição desejada.

HorizontalAling

Define o alinhamento horizontal. As opções possíveis são:

- Esquerda
- Centro
- Direita

VerticalAling

Define o alinhamento vertical. As opções possíveis são:

- Topo
- Centro
- Inferior

Margins

Define as margens da tela dentro do DockPanel. Todas as medidas da tela no projeto (tamanho, largura, etc.) são em unidades WPF (Windows Presentation Foundation).

Resources

Esta guia lista os recursos das telas. Para habilitar uma execução distribuída para as telas em Runtime e Clientes Web (ao usar imagens em telas e relatórios), importe a(s) imagem(ns) para os recursos do projeto através de [Import images...] em vez de selecionar um nome de arquivo. Um benefício adicional da guia Recursos é que quando a imagem é substituída no ResourceDictionary (o Nome do Recurso não é alterado) todas as referências a este recurso nas Telas e Relatórios são automaticamente atualizadas. A Figura 6-47 ilustra as três etapas do menu de edição de recursos.

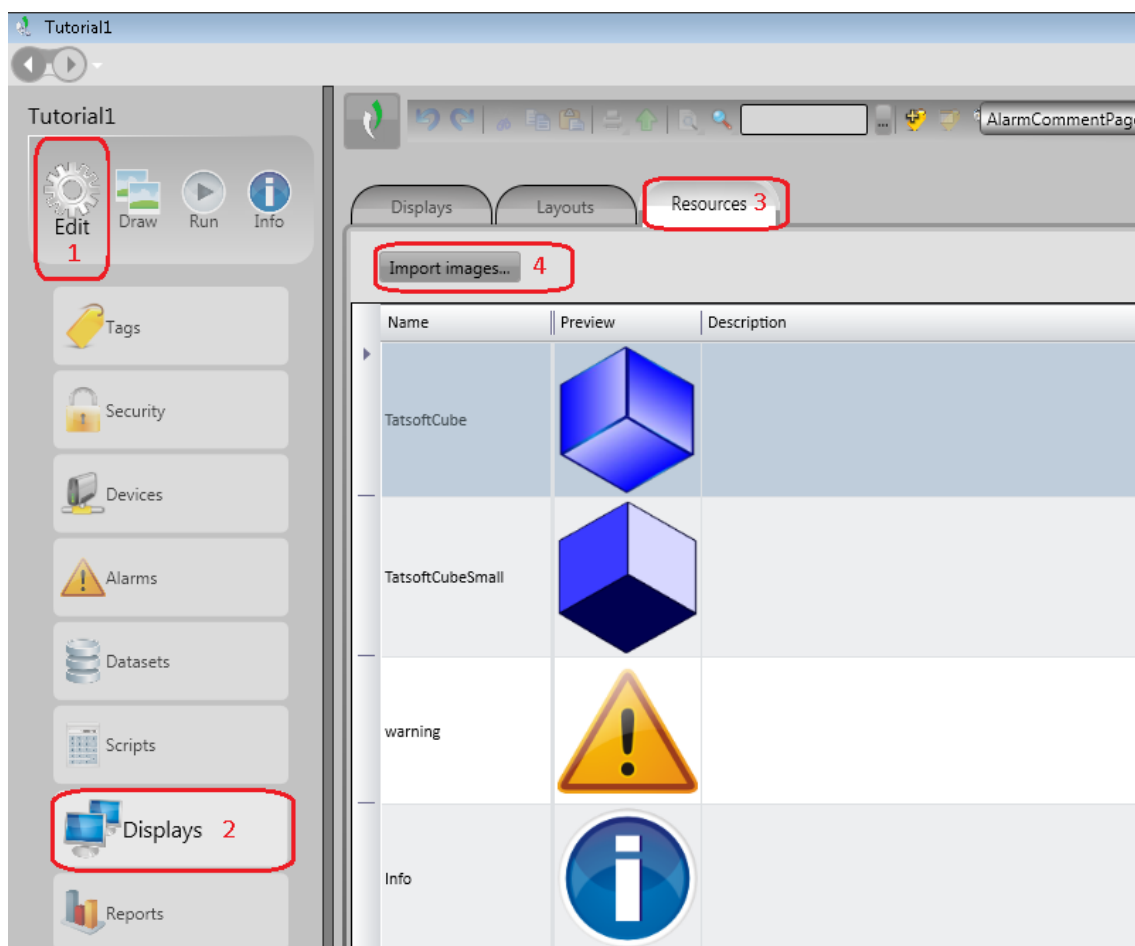



Figura 6-47. Edição de Recursos

Os passos para importar novas imagens estão ilustrados na Figura 6 47 e são descritos a seguir:

1. Selecionar o menu Edit
2. Acessar o item Displays
3. Clicar na guia Resources para acesso a todas as imagens já importadas para o projeto
4. Clicar no botão Import Images... para selecionar e importar a imagem

Name

Nome para o objeto Recurso. Clique no campo da tabela para alterar o nome do objeto.

Preview

Visualização da imagem do Recurso.

Editando Relatórios

O editor de relatório permite a inclusão de texto dinâmico, símbolo dinâmico, gráfico, conjunto de dados e consulta de resultados em um editor completo e de fácil utilização. A configuração da guia Reports está ilustrada na Figura 6-48.

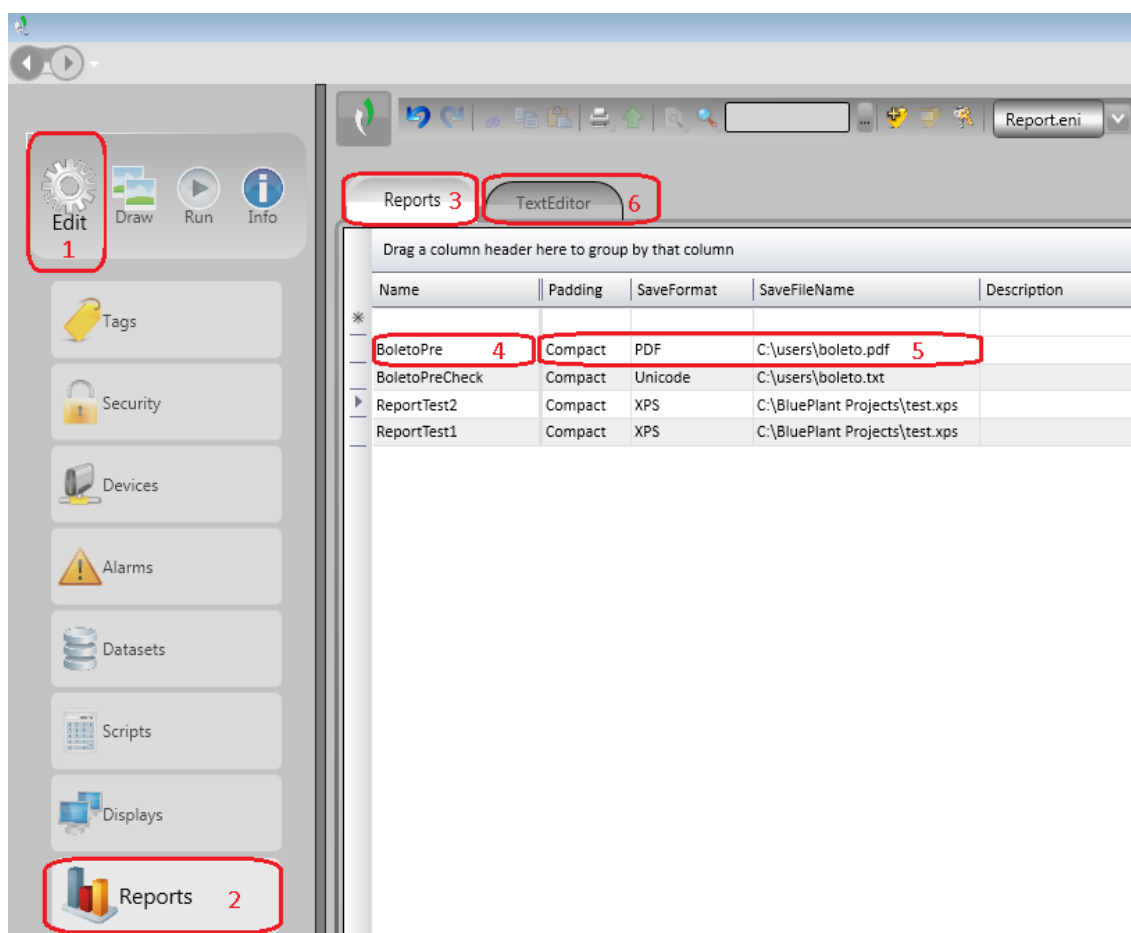



Figura 6-48. Configuração de Relatórios

Os passos para editar ou incluir um novo relatório estão ilustrados na Figura 6-48 e são descritos a seguir:

1. Selecionar o menu Edit
2. Acessar o item Reports
3. Clicar na guia Reports para acesso a todos os relatórios previamente criados no projeto
4. Incluir ou editar o nome do relatório na linha marcada com um asterisco (*)
5. Configurar as características do relatório como alinhamento dos dados, formato e nome do arquivo no qual será gerado o relatório
6. Clicar na guia TextEditor para editar o formato do relatório. A Figura 6-49 mostra com maiores detalhes

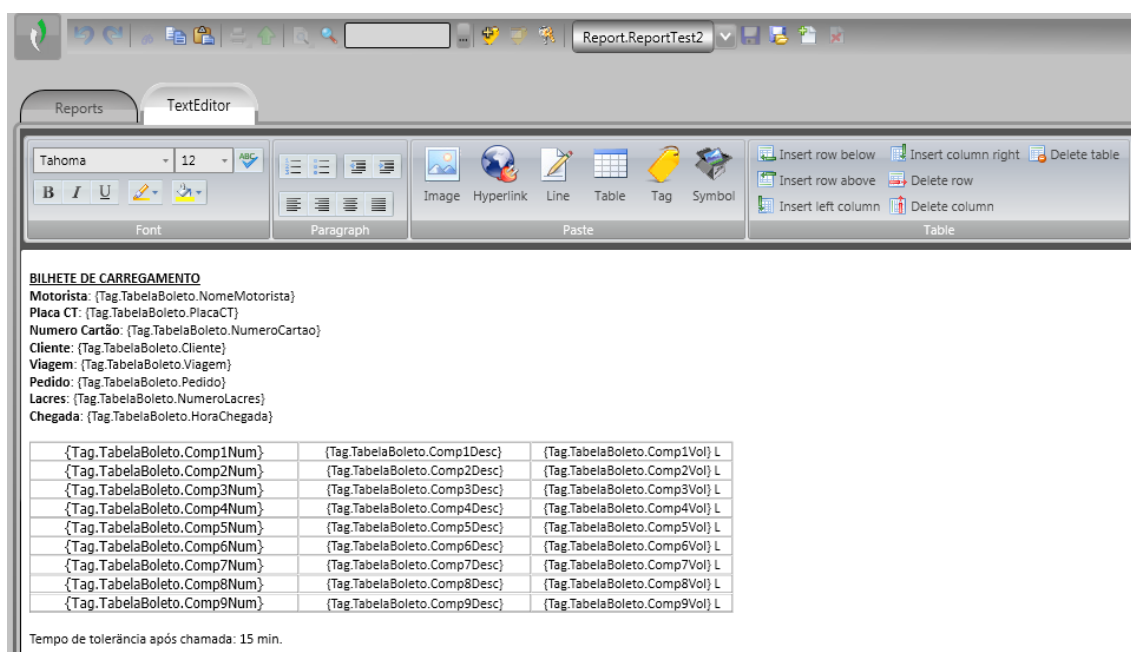



Figura 6-49. Edição de Relatório

Name

Indica o nome do relatório a ser gerado.

Padding

Define os ajustes ao substituir um TagName por seus valores. As opções possíveis são: Compact (compacto), PadRight (ajuste à direita) e PadLeft (ajuste à esquerda). O ajuste usa o número de valor exato configurado no modelo do relatório ao criá-lo no Runtime. O valor do alinhamento à esquerda ou à direita dentro do espaço está também incluído.

SaveFormat

Define o formato do arquivo usado ao salvar o relatório. As opções possíveis são: XPS, HTML, Unicode, PDF e ASCII.

Save File Name

Define o nome do arquivo usado e onde será salvo o relatório. Este campo pode incluir os valores dos tags avaliados quando da geração de um alarme, por exemplo. Para tanto, acrescente o nome do Tag nesta célula da tabela usando a notação de associação XAML entre chaves. Exemplo:

C:\MYREPORT.XPS

C:\MYREPORT-MONTH{SERVER.MONTH}-DAY{SERVER.DAY}.XPS

TextEditor

Esta guia contempla um editor de texto para criação de relatórios conforme ilustrado na Figura 6-50.

Figura 6-50. Editor de Texto

A Figura 6-51 mostra a edição e salvamento de um relatório com o auxílio do editor de texto embutido no software.

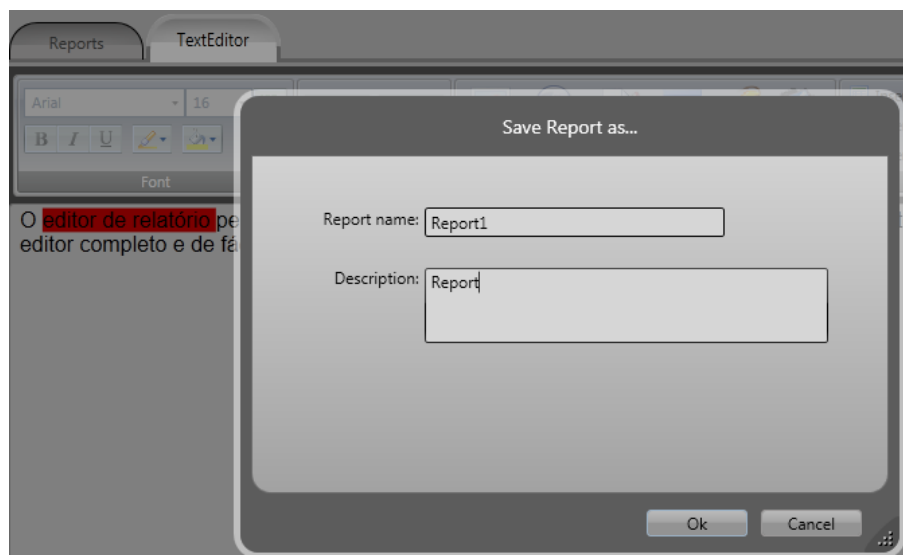



Figura 6-51. Edição e Salvamento no Editor de Texto

A Figura 6-52 mostra a consolidação do formulário editado na figura anterior com algumas configurações disponíveis no menu de edição de relatórios.

Name	Padding	SaveFormat	SaveFileName	Description
Report1	PadLeft	PDF	Test_Report	Report

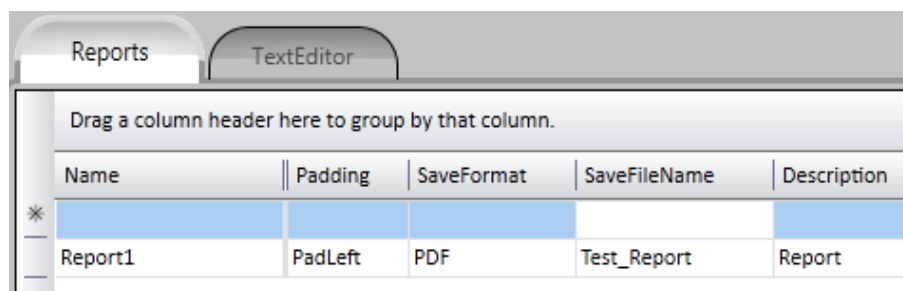
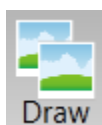



Figura 6-52. Configuração de um Relatório

Menu Draw

O ambiente de telas fornece acesso a todas as ferramentas de diagramação para compilar as telas da aplicação. Este ambiente inclui as seguintes opções: tela, códigos e símbolos. A opção Draw permite o desenho de Telas e Símbolos. Use a barra de ferramentas vertical na tela para selecionar um componente ou use a ferramenta de seleção para mover, agrupar e redimensionar componentes. A Figura 6-53 mostra a barra de ferramentas vertical.

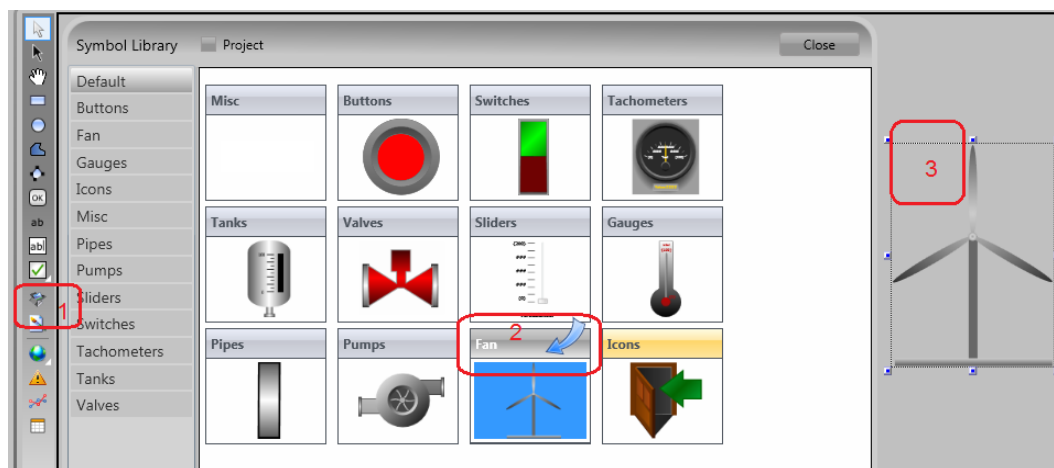



Figura 6-53. Menu Draw e Barra de Ferramentas Vertical

Os passos para a inclusão de um símbolo na tela estão numerados na figura anterior e são descritos na sequência:

1. Selecionar o símbolo a ser incluído, clicando uma vez no desenho correspondente na barra de ferramentas vertical. O símbolo será destacado
2. Clicar e arrastar o símbolo para a tela
3. Soltar o botão esquerdo do mouse para finalizar a inclusão do símbolo

A barra de ferramentas horizontal (localizada na parte inferior da tela) contém comandos para agrupar, combinar, alinhar e bloquear o(s) símbolo(s) selecionado(s). A seguir será descrito o uso da barra de ferramenta horizontal, bem como alguns comandos do menu de contexto conforme ilustra a Figura 6-54.

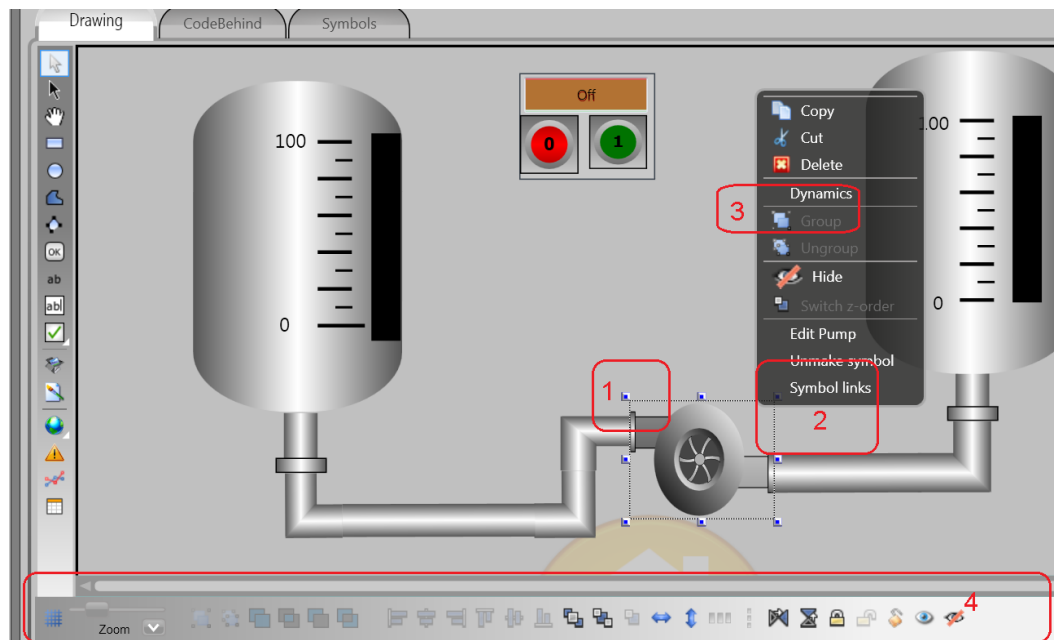



Figura 6-54. Menu Draw e Barra de Ferramenta Horizontal

A sequência de passos para a configuração das propriedades de um símbolo é:

1. Selecionar símbolo com o botão esquerdo do mouse. Para selecionar múltiplos símbolos use SHIFT+CLICAR COM BOTÃO ESQUERDO em cada componente desejado
2. Clicar no símbolo com o botão direito do mouse para abrir o menu de contexto
3. Selecionar a opção desejada no menu de contexto
4. Aplicar as propriedades desejadas para o(s) símbolo(s) usando a barra de ferramentas

Os itens que compõem a Barra de Ferramentas Vertical ilustrada na Figura 6-55 estão detalhados na sequência.

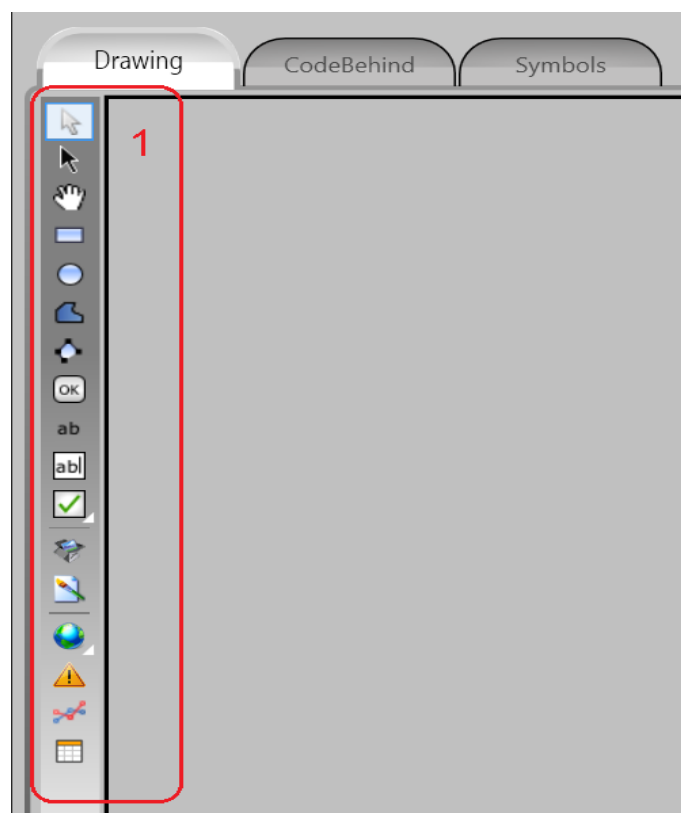



Figura 6-55. Barra de Ferramentas Vertical

Ferramentas de Seleção

Use as ferramentas de seleção para selecionar um objeto e modificar a área de visualização do desenho.

 Ferramenta de Seleção

Clique uma vez no objeto para selecioná-lo. A tecla CTRL+CLIQUE DO MOUSE permite a seleção de vários objetos ao mesmo tempo, assim como grupos de objetos. Mantenha pressionada a tecla CTRL enquanto clica em cada objeto. Para alternar a visualização entre os vários objetos selecionados utilize a tecla SHIFT+CLIQUE DO MOUSE. Clique em uma área aberta da tela e selecione um grupo de elementos, destacando os elementos desejados enquanto mantém pressionado o botão esquerdo do mouse. Clique duas vezes em um objeto para abrir a janela de configuração dinâmica que fornece as configurações para as propriedades do objeto dinâmico.

 Ferramenta de Seleção Direta

Use esta ferramenta para selecionar um objeto dentro de um grupo e modificar suas propriedades. Clique uma vez no objeto para selecioná-lo. É possível adicionar, remover e modificar os pontos em

uma Polyline através desta ferramenta. Para mover o ponto, selecione-o com um clique e pressione o botão esquerdo do mouse. Arraste o ponto para a sua nova posição. Clique duas vezes em um ponto para adicionar um novo ponto adjacente ao ponto selecionado. Clique no botão direito do mouse em um ponto para excluir o ponto selecionado.

Ferramenta Mão

Use a ferramenta de mão para modificar a janela de exibição. Clique uma vez na tela de fundo e, mantendo pressionado o botão esquerdo do mouse, mude a tela para a posição desejada.

Objetos Geométricos.

Cria um objeto Retângulo.

Cria um objeto Elipse.

Cria um objeto Polígono.

Cria um objeto Polyline.

Clique com o botão direito do mouse para finalizar o uso de cada ferramenta.

Para mais informações de como adicionar, modificar e remover pontos após a criação de um polígono ou polyline está descrito na seção Ferramentas de Seleção.

Componentes de Tela

Estes objetos criam componentes de telas.

Cria um objeto Botão.

Cria um objeto Checkbox.

Dê um clique com o botão direito no ícone do componente para acessar as ferramentas em um menu popup horizontal. Após a ferramenta ter sido selecionada no menu horizontal, ela torna-se a ferramenta padrão para aquela posição na barra vertical.

Cria um objeto RadioButton.

Cria uma ComboBox.

Cria uma ListBox.

Cria uma PasswordBox.

Cria um Controle DatePicker.

Cria uma DateTimeTextBox.

Entrada e Saída de Texto

Use as ferramentas de texto para criar objetos de entrada ou saída de texto.

ab Cria um objeto TextOutput (TextBlock) conforme ilustrado na Figura 6-56.

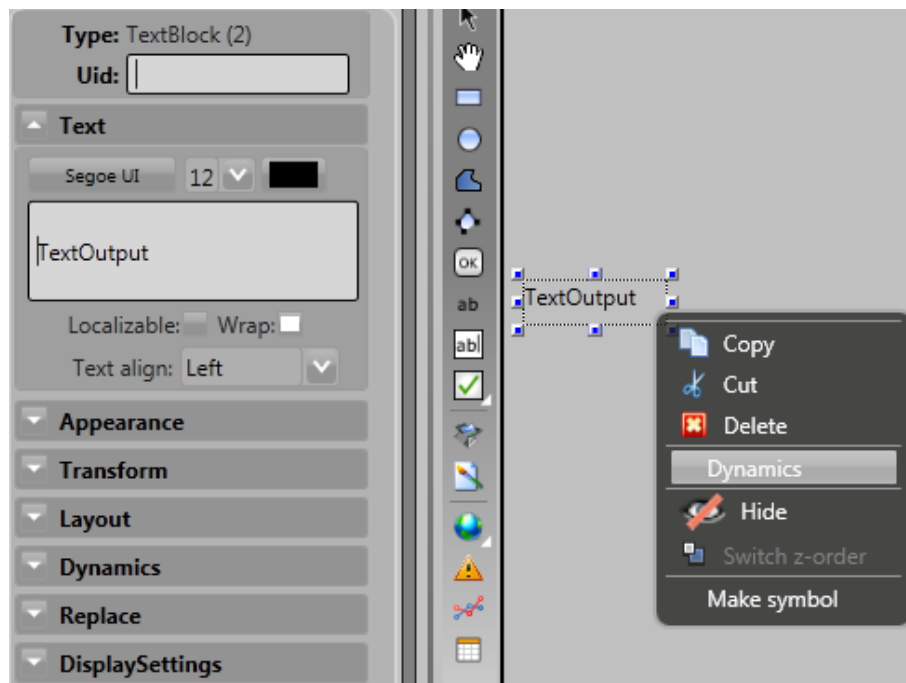



Figura 6-56. Objeto TextOutput

Um clique com o botão direito do mouse no objeto permite acessar o menu suspenso de edição associado à saída de texto, incluindo sua Configuração Dinâmica que também pode ser acessada através de um duplo clique no objeto.

abl Cria um objeto TextIO (TextBox object) conforme mostrado na Figura 6-57.

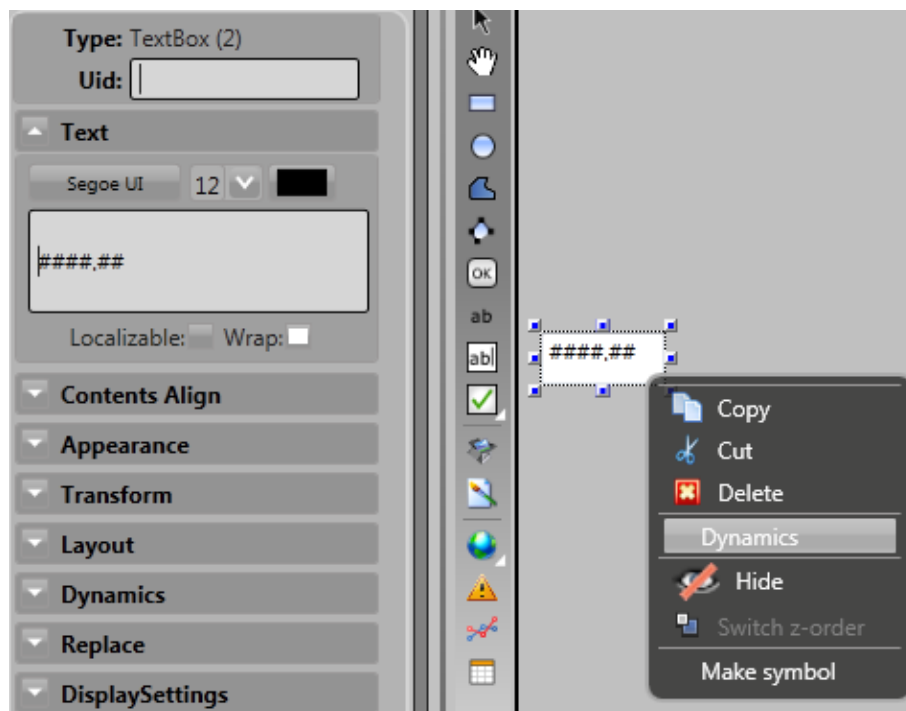



Figura 6-57. Objeto TextIO

Um clique com o botão direito do mouse no objeto permite acessar o menu suspenso de edição associado à entrada/saída de texto, incluindo sua Configuração Dinâmica que também pode ser

acessada através de um duplo clique no objeto. Por exemplo, para vincular o TextIO com o Tag, dê um clique duplo no objeto TextIO e, na janela de configuração dinâmica, escolha a dinâmica TextIO.

Biblioteca de Símbolos

Abre a biblioteca de símbolos.

 Esta biblioteca inclui símbolos incorporados e definidos pelo usuário.

Controles Avançados

 Cria um objeto WebBrowser.

Dê um clique com o botão direito no ícone do componente para acessar as ferramentas em um menu popup horizontal. Após a ferramenta ter sido selecionada no menu horizontal, ela torna-se a ferramenta padrão para aquela posição na barra vertical.

 Cria um objeto PageSelector.

 Cria um objeto ReportViewer.

Cria um objeto XpsViewer.

Cria um objeto CircularPanel.

Cria um objeto Calculadora.

Inclui um componente controle WPF.

Alarme

Objeto alarme para controlar advertências.

 Cria uma Janela de Alarme.

Este item permite criar e posicionar a janela de alarme que tem a aparência mostrada na Figura 6-58.

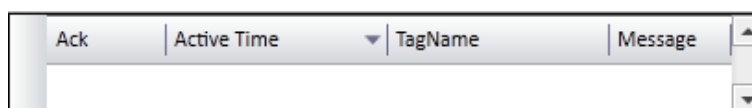



Figura 6-58. Janela de Alarmes

Figura 6-59. Configuração da Janela de Alarme

Control Name: Nome qualquer dado ao controle. É utilizado via CodeBehind para relacionar o objeto gráfico a uma classe .Net. (TAlarmWindow)

MaxLines: Número máximo de linhas que será apresentado no objeto. O filtro é feito no servidor, portanto, os dados nem chegam a ser enviados do servidor para o cliente com a finalidade de ganhar desempenho.

List: Tipos de alarmes que serão apresentados no objeto.

- OnlineAlarms: Somente alarmes online
- AlarmsHistory: Somente alarmes históricos
- Events: Somente eventos
- AlarmsHistory+Events: alarmes históricos mais os eventos

History Interval: DateTimeOffset representando o início do período de histórico.

History Interval (To): DateTimeOffset representando o fim do período de histórico.

Ack by Page: Configuração de um objeto que quando recebe um trigger reconhece todos os alarmes visíveis no objeto de alarme.

Show Column Titles: Habilita mostrar os títulos das colunas.

Show group control: Habilita a área onde pode arrastar os nomes das colunas para fazer agrupamentos.

Show group control (Label): Texto que aparecerá dentro da área do "Group control:"

AllowSort: Habilita a funcionalidade de sort no lado direito de cada coluna.

Allow Column Reorder: Habilita a funcionalidade de arrastar a coluna para o lugar de outra coluna trocando assim a ordem entre elas.

Display Value Column as String: Habilita substituir na coluna "Value" o valor pela sua string correspondente, configurado no dicionário.

Filter: Este campo é um "where" de um sql statement, portanto, a sintaxe válida é qualquer coisa válida em um where sql, levando em conta os nomes das colunas existente no objeto.Ex:

```
[TagName] = 'Tag.Tag1' AND [Group] = 'Alarm.Group.Critical'
```

Refresh: Configuração de um objeto que quando recebe um trigger faz um refresh do objeto de alarme.

Display Millisecond: Habilita a visualização dos milissegundos na formatação das datas.

Ack Selected Line: Configura o hotkey ou combinação de hotkeys para reconhecer a linha selecionada no objeto de alarme.

Lista de Columns: Através das setas para cima e para baixo, podemos mudar a ordem na qual as colunas serão visualizadas no objeto de alarme.

Para cada coluna podemos configurar:

Visible: Habilita se a coluna estará visível ou não.

Allow Filter: Habilita a opção de Filtro na coluna.

Show in column chooser: Permite selecionar e arrastar a coluna.

Title: Configura o título que será visualizado no cabeçalho da coluna.

Width: Configura a largura da coluna.

Sort: Configura o tipo de sort sue será aplicado automaticamente na coluna.

Tendência

O objeto Tendência mostra o registro do historiador de um tag.

Cria uma janela de tendência.

Posicione a janela de tendência e com um duplo clique abra a janela de configuração. Campos associados:

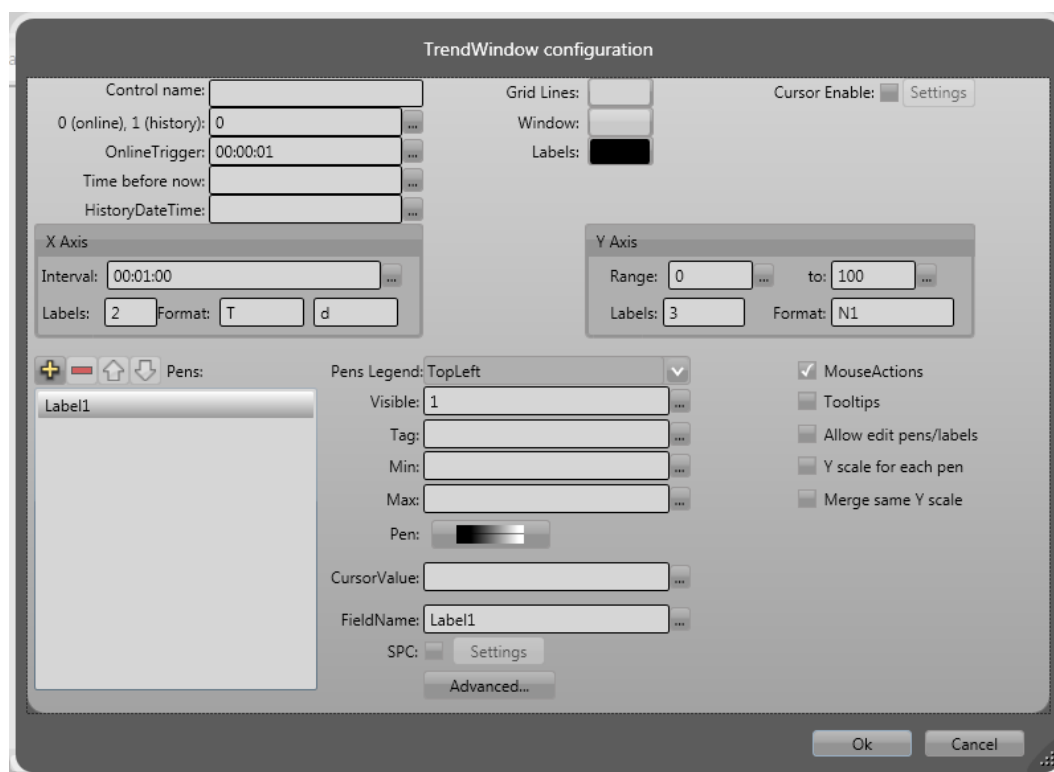



Figura 6-60. Configuração da Janela de Tendência

ControlName: Define um nome para o controle, desta forma pode ser acessado no script código associado. Consulte: Acesso ao objeto Tela no script associado.

0(online) 1(history): 0 - o controle mostra os dados online. 1 - o controle mostra os dados do historiador. Ex:

```
Tag.onLineHist
```

OnlineTrigger: Taxa de atualização do controle de tendência online.Ex:

```
00:00:01 ou {Tag.trendTimeSpan}
```

HistoryDateTime: O ponto inicial para o controle de tendência do historiador. Ex:

```
{Tag.initialTrendHistory}
```

Max Samples: Define a quantidade máxima de amostras a serem coletadas do banco de dados

Window: Define a cor da tela de fundo do controle de tendência

Labels: Define a cor dos rótulos do controle de tendência

Cursor Enable: Habilita (selecionado) ou desabilita (não selecionado) o cursor vertical

CursorPosition (%): Indica a posição do cursor, onde 0 significa posição inicial e 100 a posição final. Ex:

```
Tag.cursorPos
```

Cursor Output: Indica o valor do eixo X para a posição atual do cursor. Ex:

```
Tag.cursorOut
```

Cursor Color: Define a cor do cursor

Legenda das canetas: Define a posição da legenda das canetas no controle de tendência.

Eixo Y

Range: Define os valores mínimo e máximo para o eixo Y.

Labels: Define a quantidade de linhas de grade horizontais.

Format: Formato dos valores no eixo Y. Para formatos numéricos válidos, consulte o item referente ao formato dos tags. Ex: N1 (número com 1 casa decimal).

Eixo X

Interval: Define o TimeSpan do eixo X.

Labels: Define a quantidade de linhas de grade horizontais.

Format: O formato do eixo X é definido por dois campos: formato da primeira e da segunda linha. Isto é especialmente útil para representar marcas de rótulo que requerem dois níveis de informação. Para formatos de data e tempo válidos consulte o item referente ao formato dos tags. Ex:

T (Tempo) para o formato da primeira linha, d (data abreviada) para o formato da segunda linha.

Canetas

Visível: Mostra (1) ou oculta (0) a caneta selecionada. Ex:

1 ou {showPen1}

Tag: Define o Tag que fornecerá o valor para a caneta.

Min: Referência de escala linear para o valor do tag, de acordo com a faixa do eixo Y.

Max: Referência de escala linear para o valor do tag, de acordo com a faixa do eixo Y.

Caneta: Define o estilo, cor e espessura da linha da caneta.

Marca: Define a marca para cada ponto na linha de tendência.

CursorValue: Define o tag que receberá o valor real do eixo Y, de acordo com a posição do cursor. Ex:

`Tag.pen1CursorValue.`

FieldName: Define o nome da caneta.

Escala do Eixo Y: Considera-se, por padrão, o mínimo como zero e o máximo como 100.

O controle de tendência permite que várias canetas sejam exibidas junto. Quando as suas canetas não estão na mesma faixa use as dicas abaixo para ajustar os dados no mesmo gráfico e assim obter uma melhor visualização. Se alguma caneta apresentar uma faixa inferior, 0 a 1, por exemplo, pode-se configurar a propriedade máxima da caneta para 1. Assim, quando o valor do real for 1, o valor 100 será exibido no gráfico (escala 100/1). Se alguma caneta apresentar uma faixa superior, 0 a 1000, por exemplo, pode-se configurar a propriedade máxima da caneta para 1000. Assim, quando o valor do real for 1000, o valor 100 será exibido no gráfico (escala 1/10).

Grade de Dados

 Cria uma janela DataGrid.

Posicione a janela DataGrid e dê um duplo clique para abrir a janela de configuração.

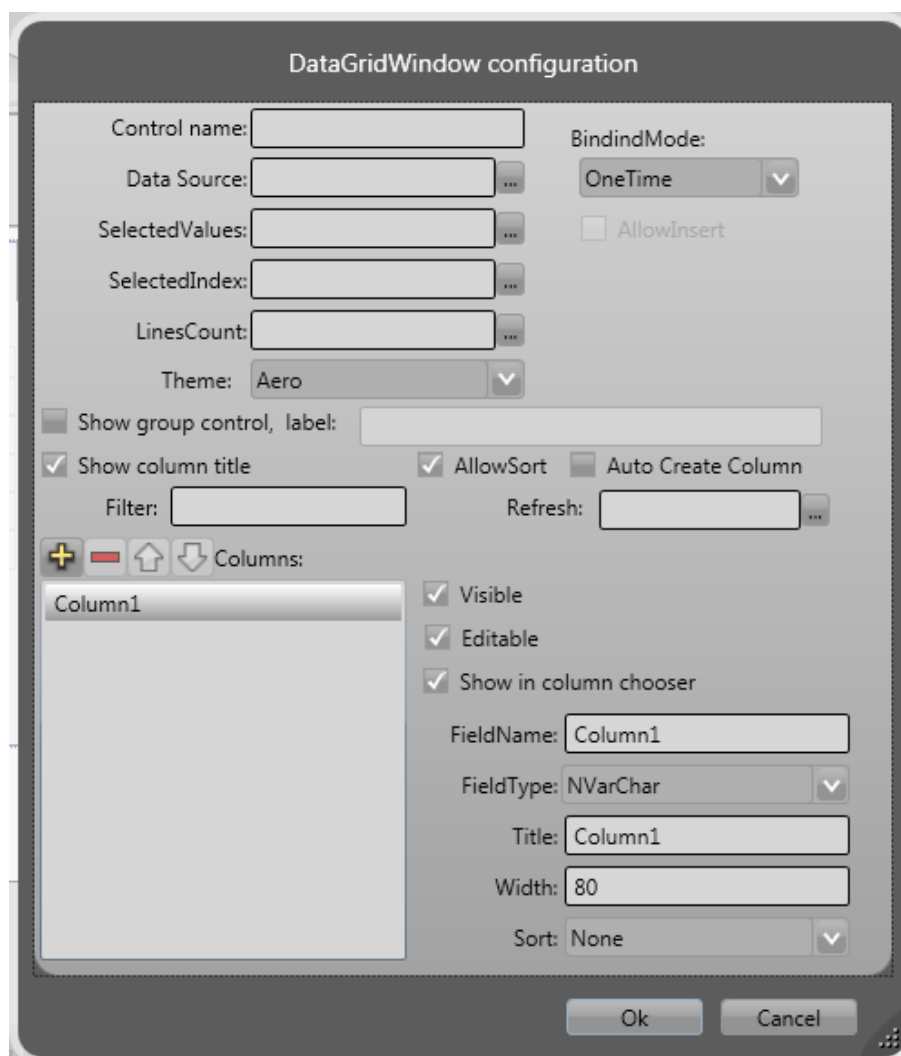



Figura 6-61. Configuração da Janela do DataGrid

Control Name: Nome qualquer dado ao controle. É utilizado via CodeBehind para relacionar o objeto gráfico a uma classe .Net. (TDataGridWindow)

Data Source: Objeto dataset no qual quer se mostrar o conteúdo no DataGrid. Pode ser tanto Dataset.Table quanto Dataset.Query.

Selected Values: Tag array do tipo Text, onde o conteúdo de cada coluna será colocado dentro de cada posição do array.

Selected Index: Objeto que recebe qual a linha que esta selecionada.

Lines Count: Objeto que recebe a quantidade de linhas existente no DataGrid.

Theme: Seleção do tema visual que será utilizado.

Binding mode: Seleciona a forma do fluxo de dados:

OneTime: O objeto grid é populado somente uma única vez.

OneWay: Os dados somente saem do banco de dados para popular o GRID.

TwoWay: Os dados saem do banco para popular o GRID, e quando modificado no GRID são atualizados também no banco de dados.

OneWayToSource: Os dados são somente atualizados do GRID para o banco de dados.

Allow Insert: Habilita inserção de novas linhas no objeto DataGrid.

Show group control: Habilita a área onde se podem arrastar os nomes das colunas para fazer agrupamentos.

Show group control (Label): Texto sue aparecerá dentro da área do "Group control:"

Show Column Titles: Habilita mostrar os títulos das colunas.

AllowSort: Habilita a funcionalidade de sort no lado direito de cada coluna.

Auto Create Column: Habilita as colunas a serem criadas automaticamente em função das colunas existentes no datatable do Dataset.Table ou Dataset.Query.

Filter: Este campo é um "where" de um sql statement, portanto, a sintaxe válida é qualquer coisa válida em um where sql, levando em conta os nomes das colunas existentes no objeto.Ex:

```
[Col1] = 'ABC' AND [Col2] = 'DEF'
```

Refresh: Configuração de um objeto que quando recebe um trigger faz um refresh do objeto GRID.

Lista de Columns: Através das setas para cima e para baixo, podemos mudar a ordem na qual as colunas serão visualizadas no objeto, e com os botões + e **-** podemos adicionar ou remover colunas.

Para cada coluna podemos configurar:

Visible: Habilita se a coluna estará visível ou não.

Editable: Habilita se o conteúdo da coluna poderá ser modificado.

Show in column chooser: Permite selecionar e arrastar a coluna.

FieldName: Nome da coluna no banco. Se esta informação estiver errada não será possível associar a coluna ao datatable do banco.

FieldType: Tipo correspondente à coluna no banco de dados.

Title: Configura o título que será visualizado no cabeçalho da coluna.

Width: Configura a largura da coluna.

Sort: Configura o tipo de sort sue que será aplicado automaticamente na coluna.

Barra de Ferramentas Horizontal

Representação Gráfica dos elementos	Descrição
	Open Grid Settings: Configuração e ajuste da grade
Zoom	Zoom: Ajuste de aproximação ou afastamento da tela
	Group: Realiza o agrupamento dos objetos selecionados
	Ungroup. Realiza o desagrupamento dos objetos selecionados
	Union: Realiza a união de objetos geométricos gerando um único objeto
	Intersect: Realiza a intersecção de objetos geométricos gerando um único objeto
	Exclude: Realiza a exclusão da forma geométrica frontal dentre os objetos selecionados gerando um novo objeto
	Exclusive-Or: Realiza a exclusão da intersecção das formas geométricas selecionadas gerando um novo objeto
	Align Left: Realiza os alinhamentos dos objetos à esquerda do último objeto selecionado
	Align Horizontal Center: Centraliza os objetos horizontalmente sendo referência o último objeto selecionado
	Aling Right: Realiza os alinhamentos dos objetos à direita do último objeto selecionado

	Aling Top: Realiza os alinhamentos dos objetos na parte superior do último objeto selecionado
	Align Vertical Center: Centraliza os objetos verticalmente sendo referência o último objeto selecionado
	Align Bottom: Realiza os alinhamentos dos objetos na parte inferior do último objeto selecionado
	Move to The Front: Movimenta para frente os objetos selecionados
	Move to The Back: Movimenta para trás os objetos selecionados
	Resize Width: Realiza o ajuste da largura dos objetos selecionados tendo como base o último objeto selecionado
	Resize Height: Realiza o ajuste da altura dos objetos selecionados tendo como base o último objeto selecionado
	Space Evenly Horizontally: Realiza o espaçamento horizontal igual entre os objetos selecionados
	Space Evenly Vertically: Realiza o espaçamento vertical igual entre os objetos selecionados
	Flip Horizontally: Realiza a inversão horizontal dos objetos selecionados
	Flip Vertically: Realiza a inversão vertical dos objetos selecionados
	Lock Element: Bloqueia todos os objetos selecionados
	Unlock Element: Desbloqueia o objeto selecionado através da Ferramenta de Seleção Direta
	Unlock All Elements Realiza o desbloqueio de todos os elementos da tela

Tabela 6-7. Elementos da Barra de Ferramentas Horizontal

Configuração Dinâmica

A opção Dynamic permite acessar a janela de Configuração Dinâmica. Selecione a dinâmica ou não a fim de habilitá-la ou desabilitá-la respectivamente. A Figura 6-62 ilustra essa configuração.

Figura 6-62. Configuração Dinâmica

Os itens que compõem a configuração dinâmica como mostrado na Figura 6-62 são descritos na sequência:

Action

Executa ações disparadas pela interface do usuário. As configurações associadas são:

- Event: escolha um dos eventos do Mouse. Mais de um evento pode ser selecionado para cada ação
    - Ex: uma ação para MouseLeftButtonDown (ao pressionar o botão esquerdo do mouse) e outra para MouseLeftButtonUp (ao soltar o botão esquerdo do mouse)
- Action: escolha uma ação para o evento determinado. Nenhum: sem ação
- SetValue: define um valor para o objeto
- Object: objeto que receberá o valor
- Value: valor que será passado ao objeto
- ToggleValue: alterna o valor do objeto. Se o valor atual do objeto é zero, o valor será 1. Se o valor atual do objeto é diferente de zero, o valor será 0. A opção “Objeto” define o objeto que será alternado
- OpenDisplay: abre uma tela. O campo Tela indica o nome da tela que será aberta
- CloseDisplay: fecha a tela
- OpenLayout: abre um Layout. Layout: o nome do Layout que será aberto
- RunScript: executa um script localizado no Código Associado à Tela. Escreva o novo nome do método e clique em Novo Botão ou selecione um dos métodos existentes na ComboBox
- RunExpressions: executa a expressão determinada
- Expression: digite a expressão. Ex: Tag.a + 1, ou Tag.a + Tag.b, ou Math.Cos(Tag.angle) * Math.PI
- Result (opcional): digite o Tag ou a propriedade que receberá o valor da expressão

Exemplos:

1. Executar Expressões:

Somar dois valores e passar o resultado para outro Tag.

Expressão:

```
Tag.quantity1 + Tag.quantity2
```

Resultado:

```
Tag.totalQuantity
```

2. Incrementar um Tag

Expressão:

```
TagCounter + 1
```

Resultado:

```
TagCounter
```

3. Incrementar um Tag (0 - 10)

Expressão:

```
If(TagCounter < 10 , TagCounter + 1 , 0)
```

Resultado:

```
TagCounter
```

Estes campos relacionados às expressões estão ilustrados na Figura 6-62.

Shine

Altera a aparência do objeto dinamicamente. As configurações associadas são:

IsMouseOver: Entre com um Tag que receberá o OverValue ou o NotOverValue.

OverValue: Valor IsMouseOver quando o mouse está sobre o objeto.

NotOverValue: Valor IsMouseOver quando o mouse NÃO está sobre o objeto.

Aparência Mouse Está Sobre: A aparência do objeto quando o mouse está sobre ele.

Opacidade: Opacidade do objeto (0 = transparente, 1 = opaco).

Escala: Tamanho do objeto (0.5 = metade, 1 = mesmo tamanho, 1.5 = uma vez e meia, 2 = tamanho duplo).

OuterGlow: Define a cor OuterGlow, a checkbox habilita ou desabilita o mesmo.

TextColor: Define a cor do texto, a checkbox habilita ou desabilita o mesmo.

Aparência Mouse Não Está Sobre: A aparência do objeto quando o mouse não está sobre ele.

Opacidade: Opacidade do objeto (0 = transparente, 1 = opaco).

Escala: Tamanho do objeto (0.5 = metade, 1 = mesmo tamanho, 1.5 = uma vez e meia, 2 = tamanho duplo).

Aparência Selecionado: A aparência do objeto quando está selecionado

IsSelected: Define se o objeto está selecionado ou não.

Opacidade: Opacidade do objeto (0 = transparente, 1 = opaco).

Escala: Tamanho do objeto (0.5 = metade, 1 = mesmo tamanho, 1.5 = uma vez e meia, 2 = tamanho duplo).

Referência de Escala: Dinâmica da escala.

o Centro

< Esquerda

^ Acima

> Direita

v Abaixo

TextIO

Dinâmica de Entrada e Saída de Texto. É composta pelas configurações ilustradas a seguir:

Campo Modo Conexão: Existem dois caminhos: entrada e saída permitidas onde, na primeira, somente a entrada é permitida (o valor do tag atual não é mostrado, mas novos valores podem ser inseridos.) e, na segunda, somente a saída é permitida.

Campo Texto: Indica o texto que será mostrado no objeto. Se o texto é um valor de Tag ou uma propriedade, ele deve estar entre chaves, por exemplo: {Tag.analogInt1}.

Campo DesignModeCaption: Valor mostrado no modo design.

- ShowObjectNames: o conteúdo do campo Texto exatamente como é.
- ShowPlaceHolders: os caracteres ### são mostrados, a quantidade de caracteres é definida pelo campo MaxLength.

Campo Faixa de Entrada: Define a faixa numérica para o valor inserido.

CampoMaxLength: Define a quantidade máxima de caracteres.

Nota:

Se o texto é um valor de Tag ou uma propriedade, ele deve estar entre chaves, por exemplo:

```
{Tag.analogInt1}
```

HyperLink

Abre um hyperlink.

HyperLink type: escolha uma das opções http, ftp, file, mailto, telnet.

Url: configure o url que será aberto.

FillColor

Altera a cor de preenchimento do objeto dinamicamente.

Expression: representa o valor usado para a dinâmica FillColor.

ChangeColor: altera a cor de preenchimento com as seguintes configurações...

- UsingLimits: a cor resultante será dada quando o valor for igual ou maior que um dos limites
- AbsoluteValue: a cor será o valor da expressão, neste caso o valor deve ser uma cor válida. Ex: "White" ou "#FFFFFFFF"

Exemplo de uso de limites:

1 - Vermelho

10 - Azul

Quando o valor for 0, o objeto terá sua própria cor (dinâmica de preenchimento de cor não terá ação)

Quando o valor for de 1 a 9, o objeto terá a cor vermelha.

Quando o valor for maior que 10, o objeto terá a cor azul.

LineColor

Altera a cor da linha do objeto dinamicamente.

Expression: Representa o valor usado para a dinâmica LineColor.

ChangeColor: Altera a cor da linha com as seguintes configurações...

- UsingLimits: a cor resultante será dada quando o valor for igual ou maior que um dos limites
- AbsoluteValue: a cor será o valor da expressão, neste caso o valor deve ser uma cor válida. Ex: "White" ou #FFFFFFFF"

Exemplo de uso de limites:

1 - Vermelho

10 - Azul

Quando o valor for 0, o objeto terá sua própria cor (dinâmica de preenchimento de cor não terá ação).

Quando o valor for de 1 a 9, o objeto terá a cor vermelha.

Quando o valor for maior que 10, o objeto terá a cor azul.

TextColor

Altera a cor do texto do objeto dinamicamente.

Expression: Representa o valor usado para a dinâmica de TextColor.

ChangeColor: Altera a cor do texto com as seguintes configurações...

- UsingLimits: a cor resultante será dada quando o valor for igual ou maior que um dos limites

- AbsoluteValue: A cor será o valor da expressão, neste caso o valor deve ser uma cor válida. Ex: "White" ou "#FFFFFFFF"

Exemplo de uso de limites:

1 - Vermelho

10 - Azul

Quando o valor for 0, o objeto terá sua própria cor (dinâmica de preenchimento de cor não terá ação)

Quando o valor for de 1 a 9, o objeto terá a cor vermelha.

Quando o valor for maior que 10, o objeto terá a cor azul.

Bargraph

Dinâmica de Gráfico de Barras.

Expression: Representa o valor usado para a dinâmica de gráfico de barras.

Faixa de valor: Os valores mínimos e máximos que corresponderão ao percentual de preenchimento mínimo e máximo.

Preenchimento (%): O percentual de preenchimento do gráfico de barras mínimo e máximo.

Cor da Barra: Cor do gráfico de barras.

Orientação: Orientação do gráfico de barras.

Representação Gráfica dos elementos	Descrição
	Movimentação das barras do gráfico da parte inferior para superior
	Movimentação das barras do gráfico do centro para as extremidades horizontais
	Movimentação das barras do gráfico da parte superior para inferior
	Movimentação das barras do gráfico da esquerda para direita
	Movimentação das barras do gráfico do centro para as extremidades verticais
	Movimentação das barras do gráfico da direita para a esquerda

Tabela 6-8. Gráfico de Barras

Visibility

Altera a visibilidade e opacidade do objeto dinamicamente.

Visível: Entre com um tag, propriedade ou expressão retornando um valor. Considere o valor resultante: zero - Objeto será visível e maior que zero - Objeto será oculto.

Dica: A string que aparecerá como dica.

Opacidade: Configurações.

- Valor do Objeto: o valor usado para definir a opacidade
- Faixa: os valores mínimos e máximos que corresponderão à opacidade mínima e máxima
- Opacidade: a opacidade mínima e máxima (0 - invisível, 0.5 - um pouco transparente, 1 - opaco)

Move and Drag

Move o objeto dinamicamente.

Modo de Associação: Configurações.

- Dois Caminhos: movimento de entrada e saída
- Somente Entrada: movimento somente de entrada, o objeto não se move quando seu valor é alterado
- Somente Saída: movimento somente de saída, o objeto não se move com a interação do usuário

Movimento Horizontal: Configurações.

- Object Value: representa o valor usado para o movimento horizontal
- Faixa: os valores mínimos e máximos que corresponderão à posição horizontal mínima e máxima
- Posição: posição horizontal mínima e máxima

Movimento Vertical: Configurações.

- Valor do Objeto: representa o valor usado para o movimento vertical
- Faixa: os valores mínimos e máximos que corresponderão à posição vertical mínima e máxima
- Posição: posição vertical mínima e máxima

Scale

Altera o tamanho do objeto dinamicamente.

Escala de largura: Configurações.

- Expressão: representa o valor usado para a escala de largura
- Faixa: os valores mínimos e máximos que corresponderão ao percentual de escala de largura mínima e máxima
- Escala (%): o percentual de escala de largura mínima e máxima

Escala de altura: Configurações.

- Expressão: representa o valor usado para a escala de altura
- Faixa: os valores mínimos e máximos que corresponderão ao percentual de escala de altura mínima e máxima
- Escala (%): o percentual de escala de altura mínima e máxima
- Referência de escala:

Centro: Escala com referência no centro do objeto.

Esquerda: Escala com referência à esquerda do objeto.

Acima: Escala com referência na parte superior do objeto.

Direita: Escala com referência à direita do objeto.

Abaixo: Escala com referência na parte inferior do objeto.

Rotate

Gira o objeto dinamicamente.

Valor do Objeto: Representa o valor usado para a rotação.

Valor do Ângulo: Os valores mínimos e máximos que corresponderão ao ângulo mínimo e máximo. Ex: 0 a 100.

Ângulo: Ângulo de rotação mínimo e máximo. Ex: 0 a 360º.

Referência de Centro:

Centro: Rotação com referência no centro do objeto.

< Esquerda: Rotação com referência à esquerda do objeto.

^ Acima: Rotação com referência na parte superior do objeto

> Direita: Rotação com referência à direita do objeto.

v Abaixo: Rotação com referência na parte inferior do objeto.

Skew

Inclina o objeto dinamicamente.

Inclinação do eixo X: configurações.

- Valor do Objeto: representa o valor usado para a inclinação do eixo X
- Faixa: os valores mínimos e máximos que corresponderão ao ângulo mínimo e máximo de inclinação do eixo X Ex: 0 a 100
- Inclinação (º): os valores mínimos e máximos de inclinação do eixo X Ex: 0 a 180º

Inclinação do eixo Y: configurações.

- Valor do Objeto: representa o valor usado para a inclinação do eixo Y
- Faixa: os valores mínimos e máximos que corresponderão ao ângulo mínimo e máximo de inclinação do eixo Y. Ex: 0 a 100
- Inclinação (º): o ângulo de inclinação do eixo Y mínimo e máximo. Ex: 0 a 180º
- Referência de Escala:

o Centro: Inclinação com referência no centro do objeto.

< Esquerda: Inclinação com referência à esquerda do objeto.

^ Acima: Inclinação com referência na parte superior do objeto.

> Direita: Inclinação com referência à direita do objeto.

v Abaixo: Inclinação com referência na parte inferior do objeto.

TextOutput

Dinâmica de Saída de Texto.

Texto: Indica o texto que será mostrado no objeto. Se o texto é um valor de Tag ou propriedade ele deve estar entre chaves. Exemplo:

`{Tag.analogInt1}.`

Localizável: Indica se e texto deve ser traduzido ao mudar o dicionário.

DesignModeCaption: O valor mostrado no modo design:

- ShowObjectNames: o conteúdo do campo Texto é mostrado exatamente como está
- ShowPlaceHolders: os caracteres ### são mostrados, a quantidade de caracteres é definida pelo campo MaxLength

MaxLength: Define a máxima quantidade de caracteres.

CodeBehind

Use DrawCodeBehind para definir um conjunto de funções vinculadas às telas conforme ilustrado na Figura 6-63.

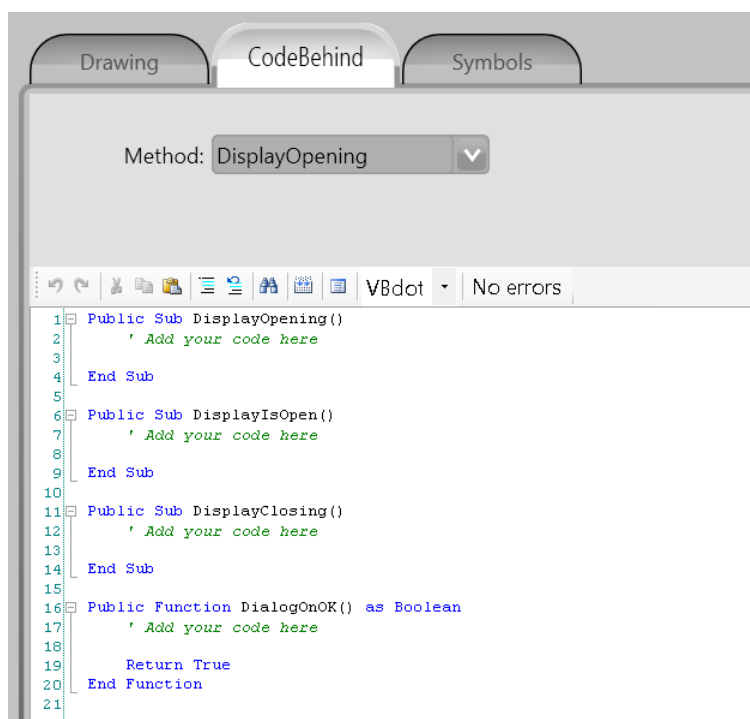



Figura 6-63. CodeBehind

Estas funções podem ser executadas na abertura ou fechamento, ou quando a tela é aberta, dependendo do código configurado. Pode-se usar o código associado para definir os métodos de uso dos comandos de entrada e de mouse a serem executados em uma tela específica.

Para telas tipo DIÁLOGO use o método DialogOnOk incorporado, o qual é chamado quando o botão OK incorporado é pressionado. Se no método retornar "TRUE", o diálogo é fechado. Se retornar "FALSE" continuará aberto. Este método é normalmente usado para garantir a validação dos dados no diálogo (o usuário é solicitado a corrigir itens incorretos antes do fechamento do diálogo). Opções:

- DisplayOpening(): executado quando a tela está abrindo
- DisplayIsOpen(): chamado em um intervalo regular enquanto a tela é aberta
- DisplayClosing(): executado quando a tela está fechando
- DialogOnOK(): chamado quando o botão OK é pressionado em uma tela de Diálogo

Retorno "1" permite o fechamento do diálogo. Retorno "0" evita que o diálogo seja fechado. Pode-se acrescentar variáveis .NET e métodos nesta página.

Nota:

Em função das telas cliente serem projetadas para execução em ambientes distribuídos e Web recomenda-se evitar o uso de funções que não permitam execução de confiança parcial ou que se refiram a caminhos de arquivo físico.

Símbolos

Símbolos são objetos definidos pelo usuário contendo desenhos e propriedades dinâmicas. A Figura 6-64 mostra um símbolo e o menu de contexto associado.

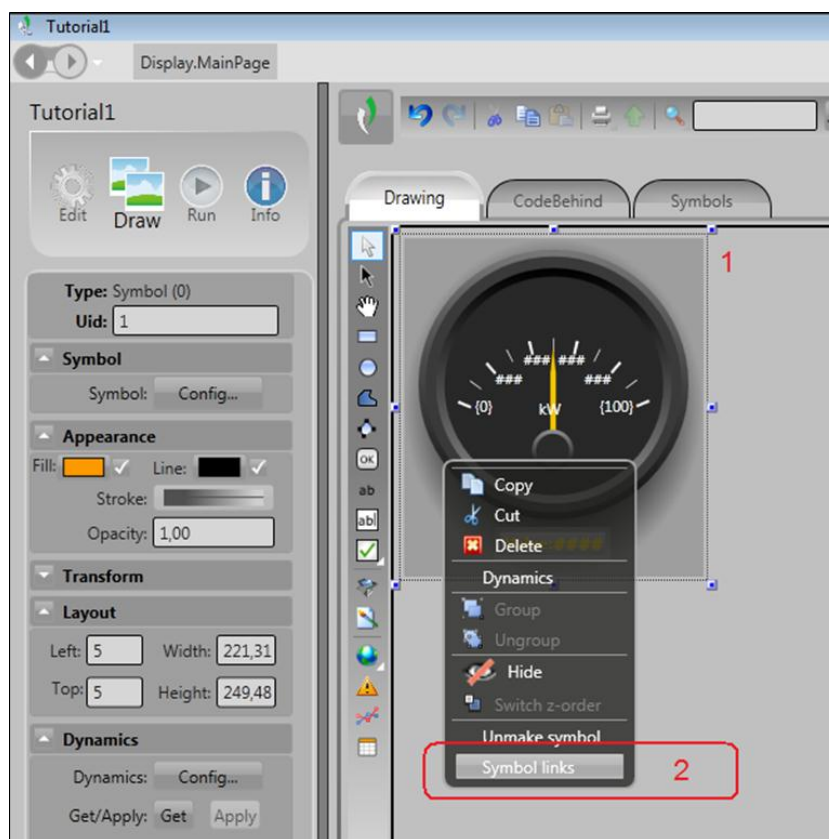



Figura 6-64. Menu Contexto do Símbolo

Para simplificar o uso de símbolos em outras telas e aplicações pode-se definir rótulos em propriedades dinâmicas onde um TagName é esperado. Use a sintaxe:

`#LabelName: ou #LabelName:DefaultValue.`

Quando usar símbolos em telas ou relatórios selecione novos nomes de tags na janela de configuração a serem aplicados nos campos de Rótulo definidos. Um conjunto de Símbolos que é visível a todos os projetos é incorporado ao BluePlant framework. Estes símbolos são definidos no arquivo SymbolLibrary.tproj localizado na pasta de instalação binária do Produto. O arquivo da biblioteca de símbolo global pode ser editado como um arquivo de projeto. Pode-se também criar e salvar seus próprios símbolos.

Criando Símbolo

As imagens da Figura 6-65, Figura 6-66 e Figura 6-67 apresentam os passos necessários para criar um símbolo. Os procedimentos também estão descritos a seguir:

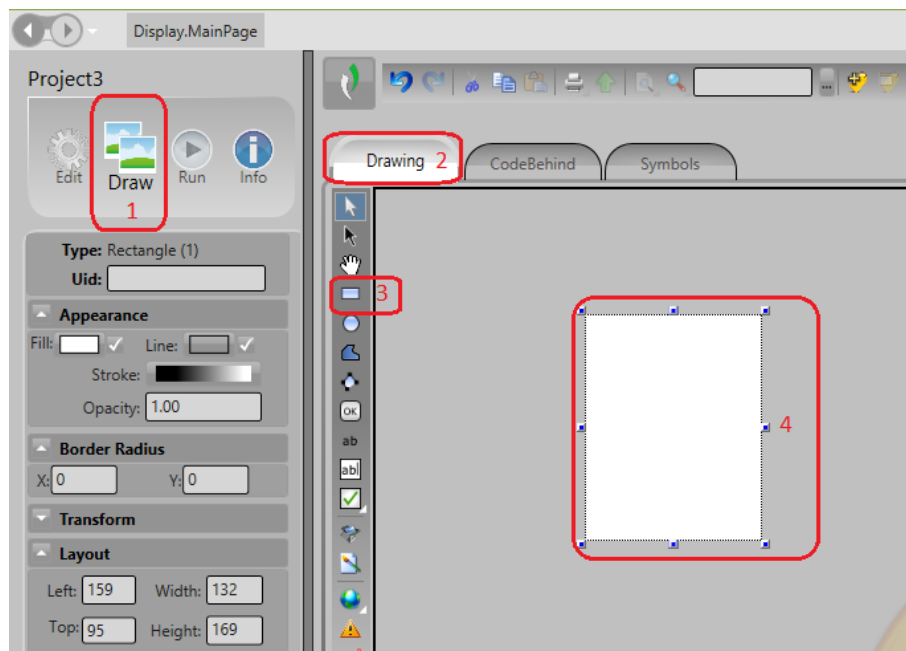



Figura 6-65. Criando um Símbolo

1. Selecionar o menu Draw
2. Clicar na guia Drawing
3. Selecionar o objeto retângulo
4. Desenhar um retângulo na tela e clicar duas vezes no objeto criado. A tela abaixo deverá aparecer

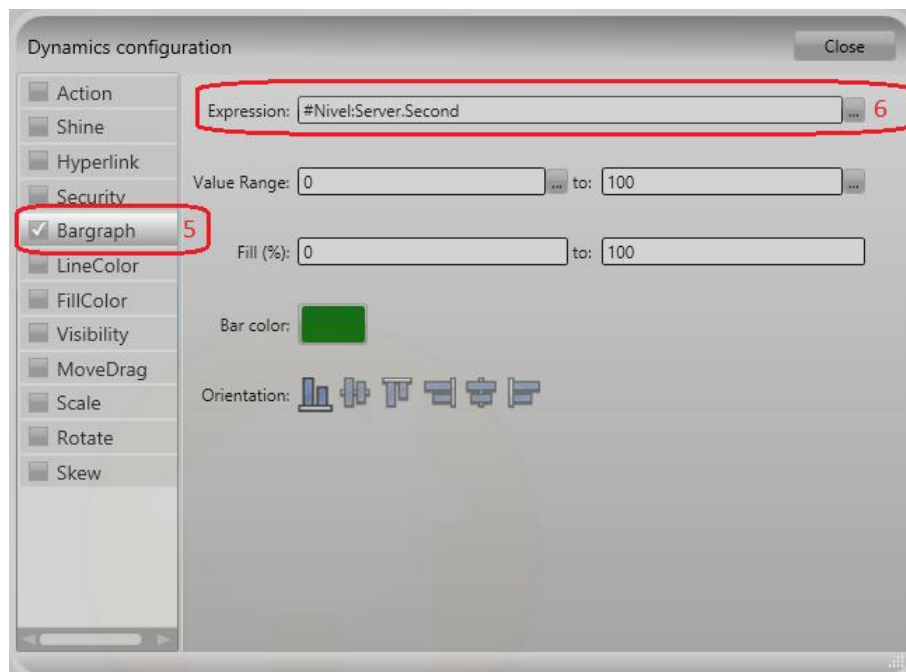



Figura 6-66. Configuração Dinâmica

5. Selecionar a opção Bargraph
6. Digitar no campo Expression a sintaxe abaixo para expor a propriedade Nível:

```
#Nivel:Server.Second
```

Nota:

As propriedades de um símbolo são expostas através do uso de mnemônicos, que são:

`#<nome>:<valor> ou #<nome>:(<expression>)`

7. Fechar a tela de Dynamics Configuration
8. Selecionar todos os objetos que compõem o símbolo, neste caso o retângulo
9. Clicar com botão direito do mouse para abertura do menu de contexto
10. Selecionar a opção "Make new Symbol"
11. Preencher as informações: Nome do símbolo, categoria, descrição e clicar em YES
12. Este símbolo estará disponível para ser inserido em qualquer ponto do projeto através do Symbol Library

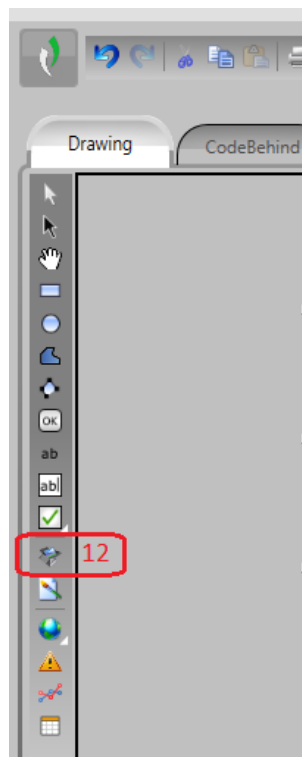



Figura 6-67. Configuração Dinâmica

13. Ir no icone Symbol Library como mostrado na figura acima e inserir o símbolo criado na tela
14. Ir na tela de configuração do símbolo dando um duplo click. Aparecerá uma propriedade "Nivel" com o valor "Server.Second" que pode ser alterado para o tag desejado, podendo cada símbolo ter o seu valor de nível independente

Modificando um Símbolo

Os passos a seguir, descrevem o que deve ser feito para que um símbolo seja modificado.

1. Selecionar o símbolo que se deseja modificar
2. Clicar com o botão direito do mouse para abertura do menu de contexto
3. Selecionar a opção "Edit <nome do símbolo>"
4. Fazer todas as alterações desejadas no símbolo
5. Selecionar todos os objetos que compõem o símbolo
6. Clicar com botão direito do mouse para abertura do menu de contexto.
7. Selecionar a opção "Update <nome do símbolo>"
8. Todos os símbolos com este nome da tela corrente serão alterados, se desejar que todos os símbolos com este nome do projeto sejam alterados será necessário fazer um build salvando as telas

Menu Info

Project

Submenu de informações do projeto atual.

Version

A guia Version, ilustrada na Figura 6-68, contém informações sobre o projeto atual. Alguns campos de informação também estão disponíveis durante o Runtime usando Namespace Info.

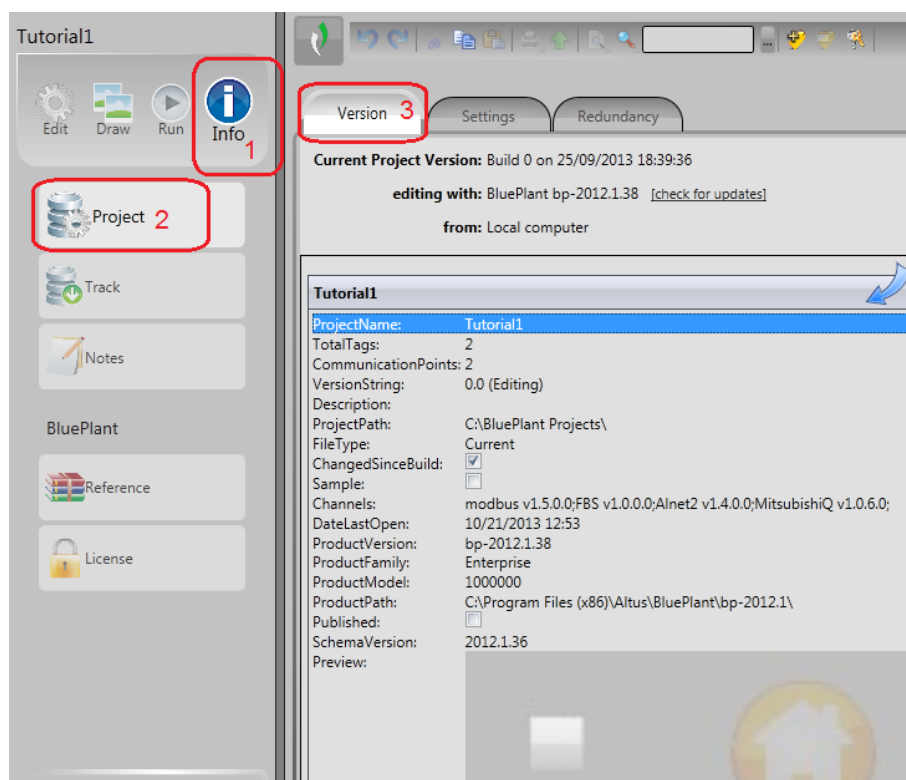



Figura 6-68. Guia Version

Os itens que compõem o menu de informação estão numerados na figura anterior e são descritos na sequência:

1. Selecionar a opção Info
2. Selecionar a opção Project
3. Selecionar a opção Version para exibição dos dados relativos ao projeto

Settings

Na sequência são explicados os itens da guia Settings conforme ilustrado na Figura 6-69.

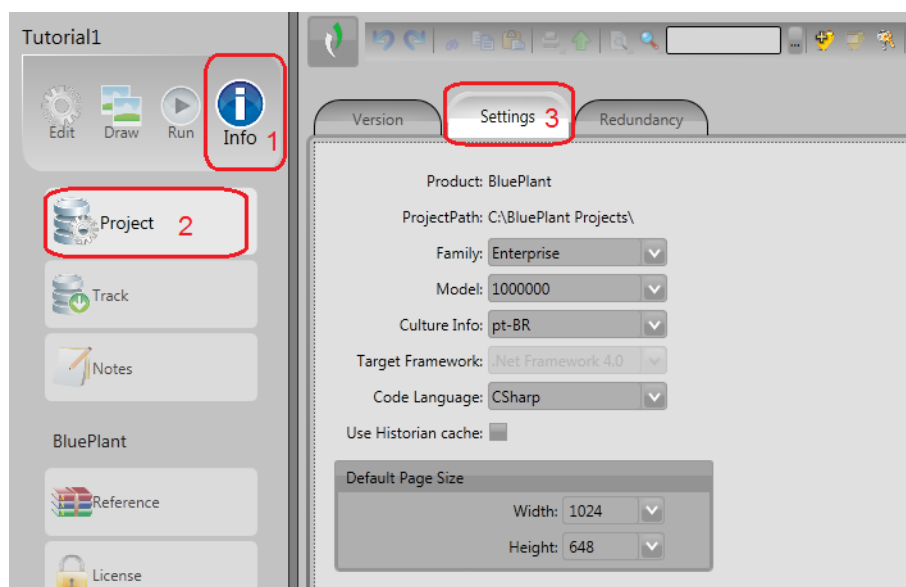



Figura 6-69. Guia Settings

Os itens que compõem a guia Settings estão numerados na figura anterior e são descritos na sequência:

1. Selecionar a opção Info
2. Selecionar a opção Project
3. Selecionar a opção Settings

Na sequência são explicados os itens do menu Info Settings conforme mostrado na Figura 6-69.

Configurações do Projeto

Define as configurações globais para o projeto. Alguns campos de informação também estão disponíveis durante o Runtime usando Namespace Info.

BluePlant Model

Seleciona o Modelo de BluePlant correspondente. As seguintes opções estão disponíveis:

- Express
- Enterprise
- Student
- Lite

Model

Seleciona o modelo de Runtime. As seguintes opções de Runtime estão disponíveis:

- 75
- 150
- 300
- 500
- 1.500
- 2.500
- 5.000
- 15.000
- 25000

- 50.000
- 100.000
- 1.000.000

Culture Info

Seleciona o idioma do projeto.

Target Framework

Seleciona o framework, plataforma o qual o BluePlant executa.

Code Language

Seleciona a linguagem do Script. As seguintes Linguagens podem ser selecionadas:

- VBdotNet
- CSharp

Default Page Size

Define o tamanho padrão da página. A Figura 6-70 ilustra essa seleção.

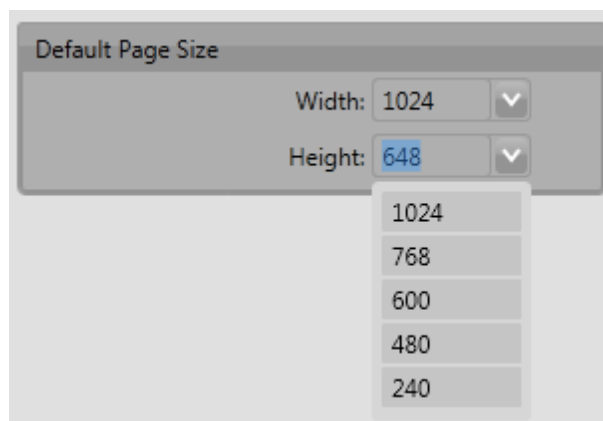



Figura 6-70. Ajuste do Tamanho da Página

Redundancy

Na sequência são explicados os itens da guia Redundancy conforme ilustrado na Figura 6-71.

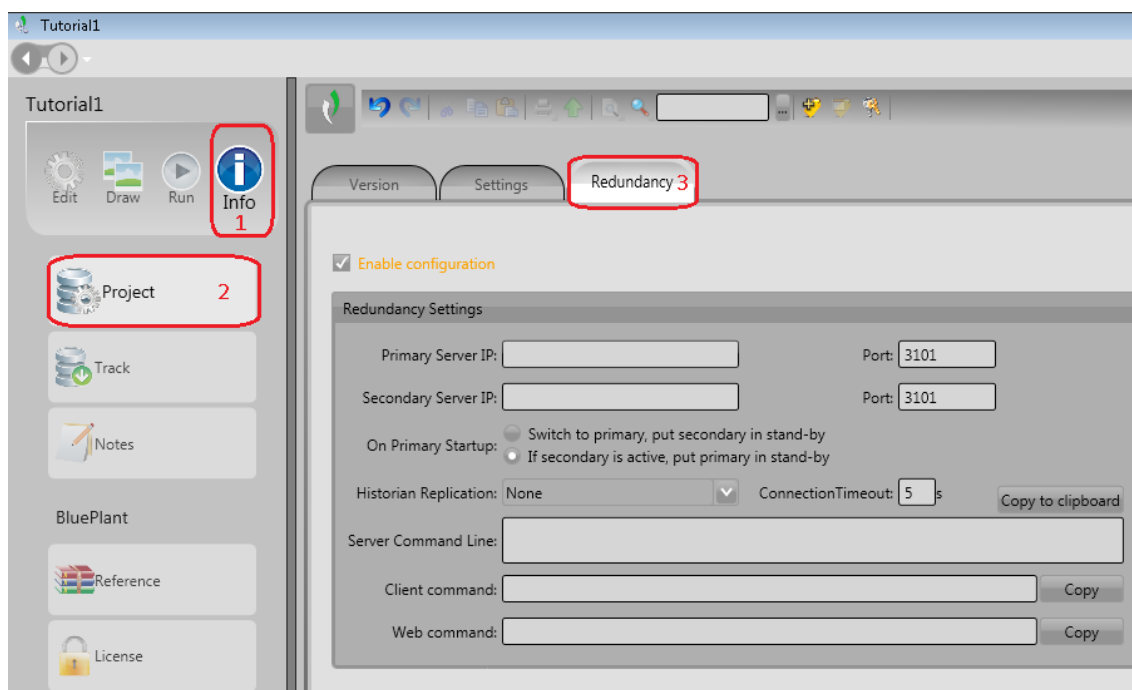



Figura 6-71. Guia Redundancy

Enable configuration: Opção para habilitar a redundância no projeto.

Primary Server IP: Campo para configuração do endereço IP do servidor denominado primário.

Secondary Server IP: Campo para configuração do endereço IP do servidor denominado secundário.

Port: Número da porta que o servidor irá utilizar para comunicar com os outros módulos e servidores.

On Primary Startup: Define o comportamento do servidor primário quando este for iniciado.

- Switch to primary, put secondary in standy-by
- If secondary is active, put primary in stand-by

Historian Replication: Define quais históricos irão fazer replicação de dados automáticos.

- None
- Alarm Historian
- Tag Historian
- Alarm and Tag Historian

Connection Timeout: Tempo especificado que o servidor reserva para aguardar quando não há comunicação entre os servidores redundantes e assume como ativo.

Server Command Line: Linha de comando utilizada para iniciar os servidores.

Client Command: Linha de comando utilizada para iniciar os clientes.

Web command: URL utilizada para acessar o projeto através do Internet Explorer.

Mais informações sobre o comportamento de servidores redundantes pode ser encontrado em Cenários de Sistemas Típicos.

Track

Tables

Esta guia do menu Info, opção Track, rastreia o status das tabelas de configuração. A Figura 6-72 mostra essa seleção.

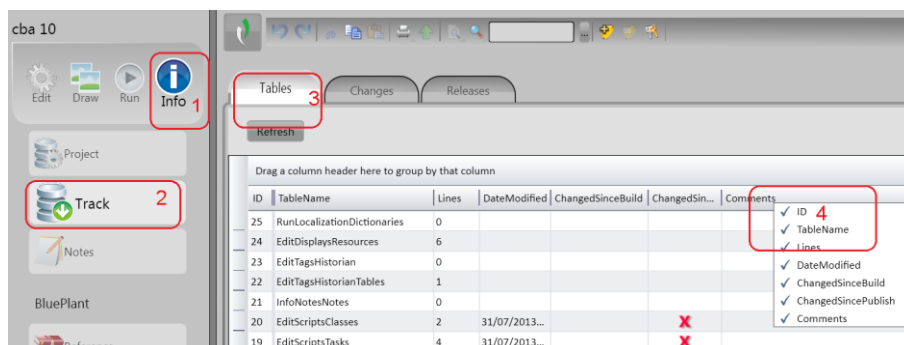



Figura 6-72. Guia Tables do Menu Info, Opção Track

Os itens que compõem o menu Info - Track - Tables estão numerados na figura anterior e são descritos na sequência:

1. Selecionar a opção Info
2. Selecionar a opção Track
3. Selecionar a opção Tables
4. Clicar com o botão direito do mouse em qualquer um dos cabeçalhos de coluna e selecionar as opções a serem exibidas

Os itens da guia Tables são explicados a seguir e podem ser visualizados na Figura 6-72.

TableName

Nome da Tabela do projeto. Atributos: ReadOnly (somente leitura).

Lines

Número atual de linhas na tabela de configuração. Atributos: ReadOnly (somente leitura).

DateModified

Data da última alteração. Atributos: ReadOnly.

ChangedSinceBuild

Alterações desde a última execução do comando de compilação. Atributos: ReadOnly (somente leitura).

ChangedSincePublish

Alterações desde a última execução do comando de publicação. Atributos: ReadOnly (somente leitura).

Comments

Comentários definidos pelo usuário. Atributos: “editável”.

Changes

Esta guia do menu Info opção Track, rastreia as modificações do projeto. A Figura 6-73 mostra essa seleção. O campo Tracking define quando as mudanças serão salvas.

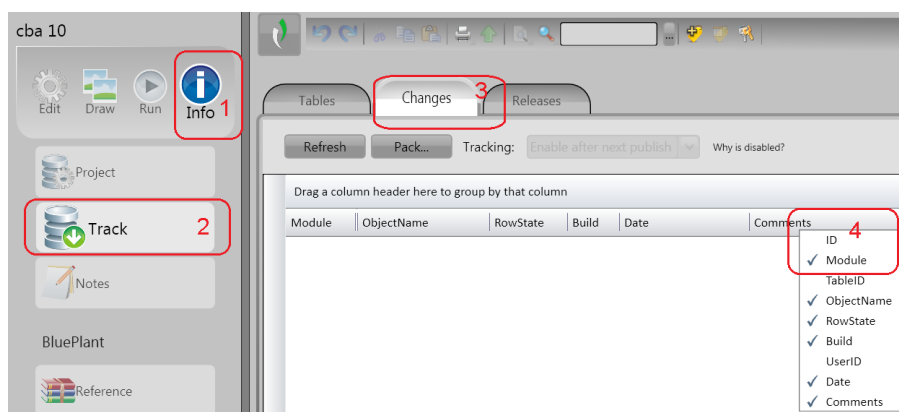



Figura 6-73. Guia Changes do Menu Info, Opção Track

Os itens que compõem o menu Info - Track - Changes estão numerados na figura anterior e são descritos abaixo:

1. Selecionar a opção Info
2. Selecionar a opção Track
3. Selecionar a opção Changes
4. Clicar com o botão direito do mouse em qualquer um dos cabeçalhos de coluna e selecionar as opções a serem exibidas

Module

Nome do módulo onde o objeto está definido. Atributos: ReadOnly (somente leitura).

ObjectName

Nome do objeto ou índice de linhas. Atributos: ReadOnly (somente leitura).

RowState

Operação executada no objeto. Atributos: ReadOnly (somente leitura).

Build

Número de compilação do projeto quando a operação foi executada. Atributos: ReadOnly (somente leitura).

Date

Data de modificação da execução. Atributos: ReadOnly (somente leitura).

Comments

Comentários definidos pelo usuário. Atributos: "editável".

Releases

Esta guia do menu Info opção Track rastreia as versões publicadas do projeto. A Figura 6-74 mostra essa seleção.

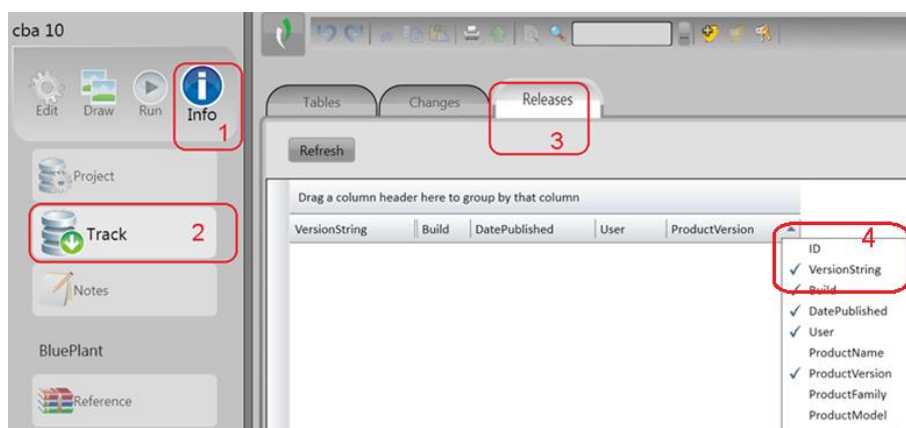



Figura 6-74. Guia Releases do Menu Info, Opção Track

Os itens que compõem o menu Info - Track - Releases estão numerados na figura anterior e são descritos a seguir:

1. Selecionar a opção Info
2. Selecionar a opção Track
3. Selecionar a opção Releases
4. Clicar com o botão direito do mouse em qualquer um dos cabeçalhos de coluna para selecionar as opções a serem exibidas

Os itens da guia Releases estão explicados abaixo e podem ser visualizados conforme Figura 6-74.

Versionstring

Versão publicada. Atributos: ReadOnly (somente leitura).

Build

Número da compilação quando a versão foi publicada. Atributos: ReadOnly (somente leitura).

DatePublished

Data de publicação da versão para este projeto. Atributos: ReadOnly (somente leitura).

User

Nome do usuário que publicou esta versão. Atributos: ReadOnly (somente leitura).

Notes

Permite a criação de notas e mensagens tipo "PostIt" visualizáveis por todos os membros da equipe de design do projeto. As notas são visíveis na área de trabalho durante a edição do projeto.

Esta interface fornece uma tabela do tipo "quadro de mensagens" para a criação dos lembretes críticos (notas) que são acessadas por todo o pessoal envolvido no desenvolvimento do projeto.

Quando o usuário clica em uma mensagem "fechada" ela permanece fechada, porém não é excluída. Para excluir uma mensagem selecione a nota na tabela, dê um duplo clique e selecione "Excluir Linha Selecionada". Um clique em "IsOpen" na linha de mensagem faz com que ela seja aberta.

Para implementar mensagens para os operadores de projeto durante a execução do mesmo use a função OpenPopupNote() disponível nas opções de objetos runtime do cliente.

Esta guia do menu Info opção Notes edita notas do usuário sobre a configuração do projeto. A Figura 6-75 mostra essa seleção.

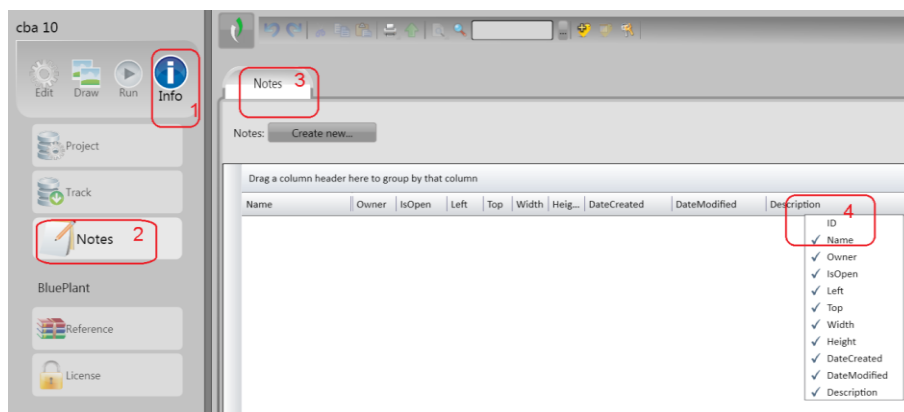



Figura 6-75. Guia Notes do Menu Info Opção Notes

Os itens que compõem o menu Info - Notes - Notes estão numerados na Figura 6-75 e são descritos a seguir:

1. Selecionar a opção Info
2. Selecionar a opção Notes
3. Selecionar a guia Notes
4. Clicar com o botão direito do mouse em qualquer um dos cabeçalhos de coluna e selecionar as opções a serem exibidas

Ferramentas da Aplicação

Esta seção explora as ferramentas disponíveis no BluePlant.

- TStartup
- PropertyWatch
- TraceWindow
- ModuleInformation

Tstartup

Ao executar o projeto usando o Gerenciador ou interfaces de boas-vindas, a janela TStartup aparece conforme ilustrado na Figura 6-76.

Conteúdo: informações sobre Status dos Módulos (rodando, parado ou pausado). Os comandos associados são:

- Botão Play: inicia a execução do módulo
- Botão Stop: interrompe a execução de um módulo, fecha as conexões e libera os recursos
- Botão Pause: pausa a execução do módulo, geralmente usado pelo recurso de redundância de servidores
- Botão Shutdown: para o servidor BluePlant (TServer) e todos os módulos
- Botão Watch: inicia o PropertyWatch
- Botão Trace: inicia o TraceWindow
- Botão Info: inicia o ModuleInformation

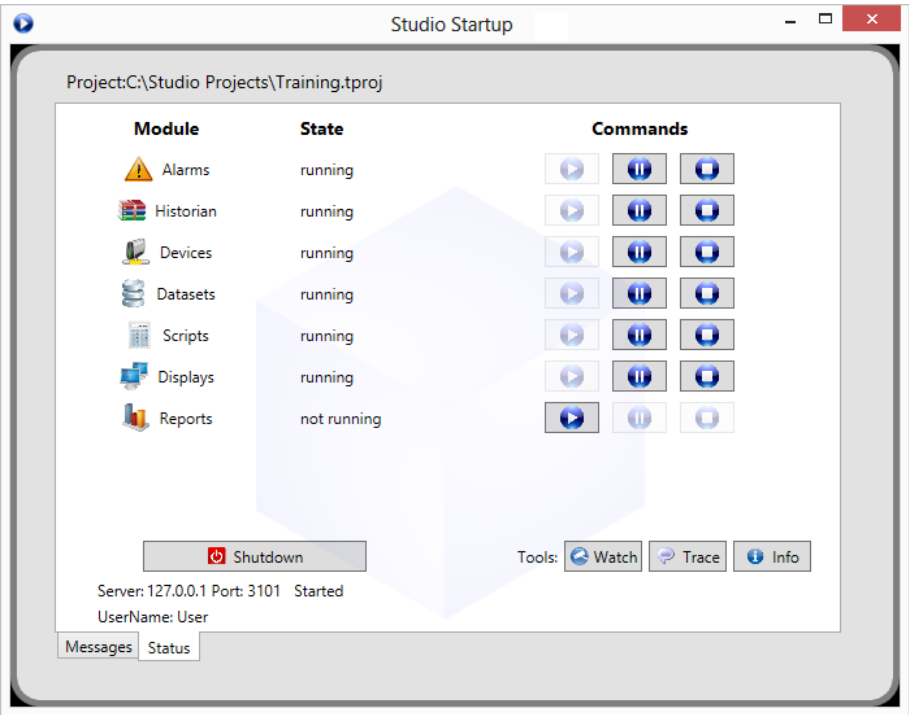



Figura 6-76. Janela de Startup

Pode-se chamar a janela de Startup usando a linha de comando. Os comandos disponíveis podem ser consultados na seção Linhas de Comando (Tstartup).

PropertyWatch

O PropertyWatch pode ser usado para acessar o servidor de Domínio de tags e Propriedades e mudar seus valores conforme ilustrado na Figura 6-77.

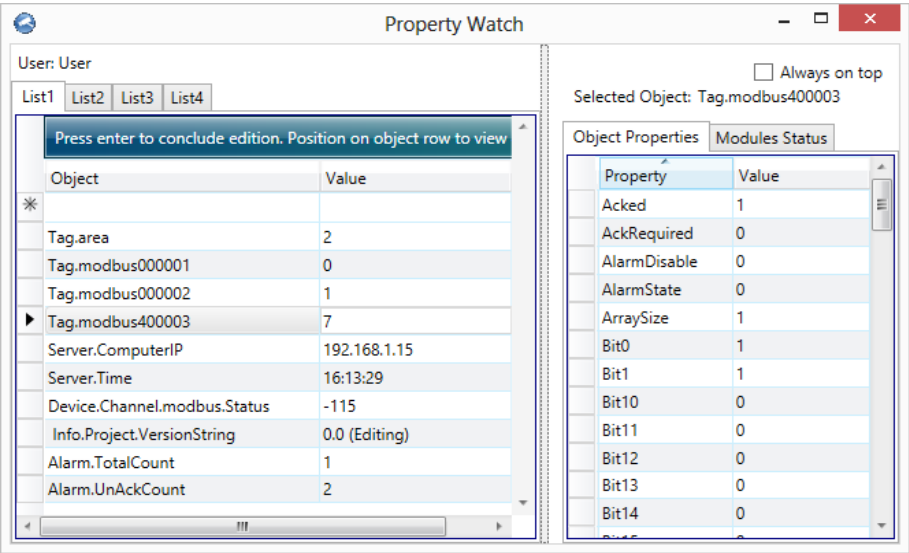



Figura 6-77. PropertyWatch

Pode-se chamar o PropertyWach usando a linha de comando. Os comandos disponíveis podem ser consultados na seção Linhas de Comando (PropertyWatch).

TraceWindow

A TraceWindow é usada para capturar as mensagens dos módulos de Runtime do BluePlant conforme ilustrado na Figura 6-78. Estas mensagens são úteis para compreender o comportamento em Runtime.

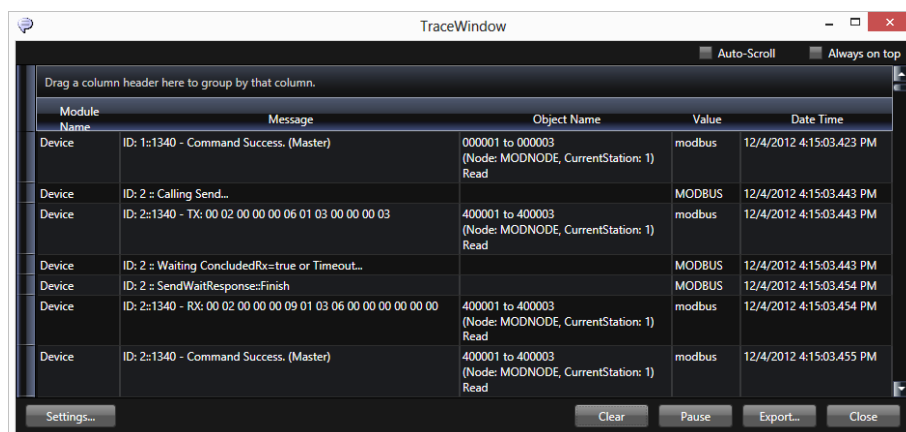



Figura 6-78. TraceWindow

Ajustes:

As opções de ajuste mostradas na Figura 6-79 são:

- Modules: determina de quais módulos a TraceWindow recebe mensagens
- Type: determina os tipos de mensagem que a TraceWindow recebe

Pode-se adicionar uma Tag ou propriedade no campo ObjectName, de maneira que, ao clicar no botão Add, uma notificação será enviada para a TraceWindow sempre que o objeto mudar o seu valor.

Saídas:

- Grid: indica o número máximo de arquivos que aparecem na grade
- File: um nome de arquivo pode ser definido para armazenamento das mensagens

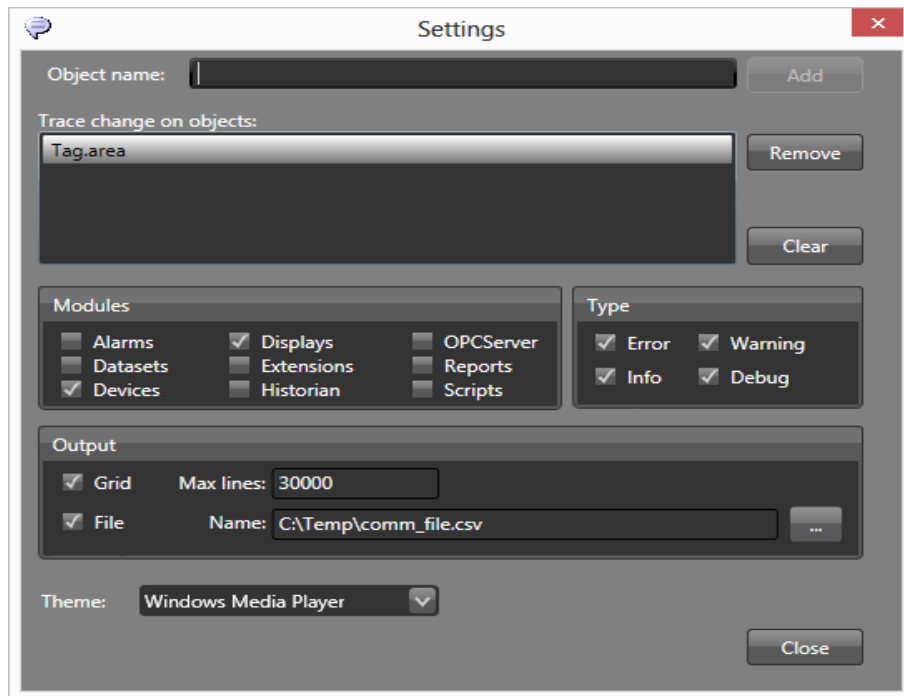



Figura 6-79. Configurações

Pode-se chamar a TraceWindow usando a linha de comando e parâmetros específicos. Para enviar mensagens para a TraceWindow do projeto, deve-se chamar a função Info.Trace(string str).

TraceWindow Message

Seleciona os tipos de mensagem da janela de rastreamento. Os seguintes tipos de mensagem da janela de rastreamento estão disponíveis:

- Erro
- Informação
- Advertência
- Depuração

ModuleInformation

ModuleInformation contém informações avançadas sobre a execução dos módulos conforme ilustrado na Figura 6-80.

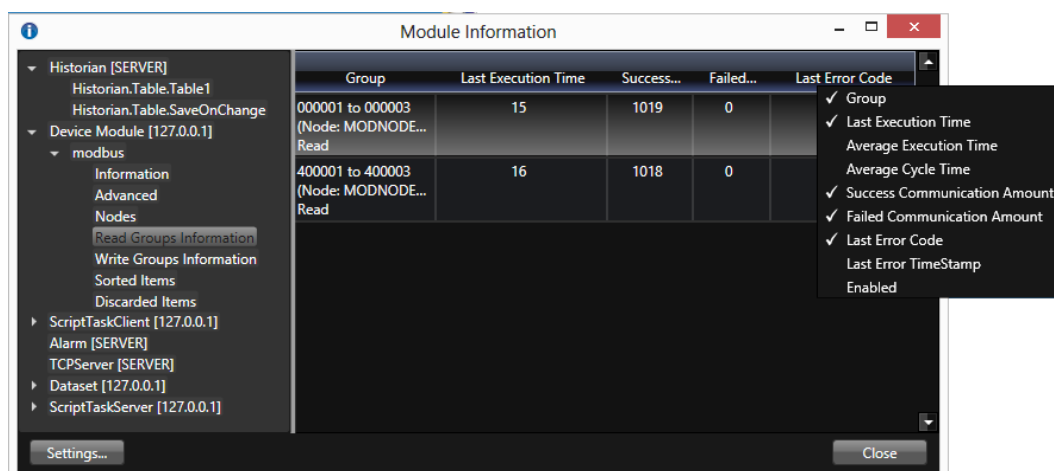



Figura 6-80. Informações do Módulo

Pode-se chamar o ModuleInformation usando a linha de comando e parâmetros específicos.

Objetos Runtime

Esta funcionalidade permite visualizar todos os elementos abertos, incluindo Telas, Scripts e Relatórios. Os objetos do Runtime estão em grupos que contem informações específicas de suas funcionalidades, sendo chamados de Namespace. Segue a lista dos Namespaces disponíveis:

- Namespace Tag
- Namespace Security
- Namespace Alarm
- Namespace Device
- Namespace Dataset
- Namespace Script
- Namespace Display
- Namespace Report
- Namespace Info
- Namespace Server
- Namespace Client

Durante a configuração do projeto, digite diretamente no campo. O "Intellisense" guiará o usuário em relação aos namespaces válidos para aquele campo assim como os membros disponíveis.

No Editor de Código (interfaces ScriptCodeEditor e DrawCodeBehind) e na dinâmica TextOutput é necessário utilizar o prefixo "@" para evitar conflito com os Namespaces.NET.

Exemplo:

Nos diálogos e grades (Figura 6-81), use:

```
Tag.Analog1
Alarm.Group.Warning.TotalActive
Device.Node.Node1.Status
```

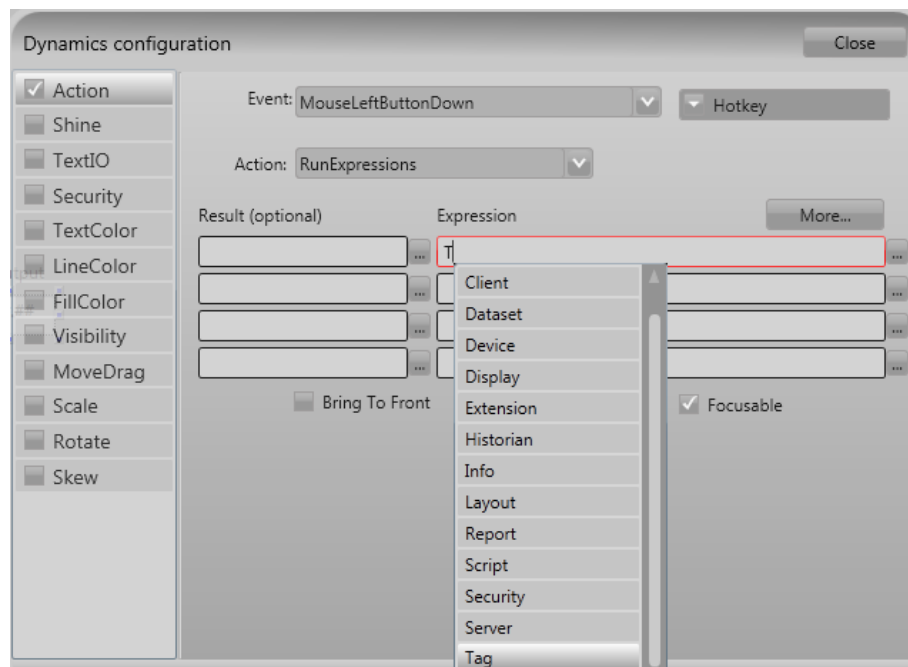



Figura 6-81. Namespaces em Diálogos

No código script (Figura 6-82), use:

```
@Tag.Analog1
@Alarm.Group.Warning.TotalActive
@Device.Node.Node1.Status
```

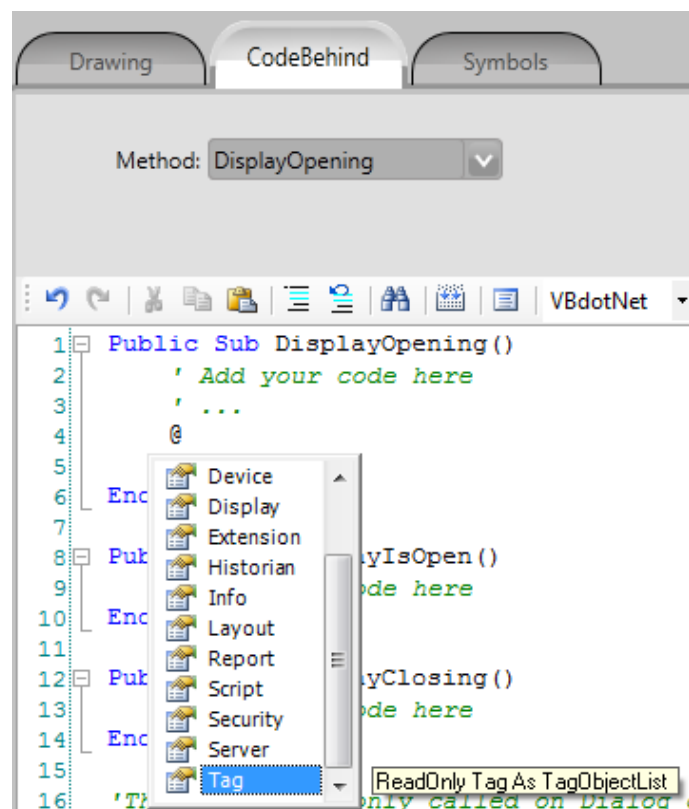



Figura 6-82. Namespaces em Scripts

Alguns campos nas Grades e Diálogos são permitidos somente para um tipo de objeto (um namespace), por exemplo, "Tag" ou "Display".

Quando o usuário digita um Namespace neste campo o Intellisense o levará até os objetos disponíveis.

Namespace Tag

Todas as variáveis de tempo real do projeto, ou "Tags de Projeto", como são normalmente chamadas nos contextos de automação de processos, estão disponíveis neste Namespace. Os tags são criados na interface EditTagsObjects. Use a sintaxe Tag.Namespace para referir-se ao tag criado e Tag.namespace.Min para acessar a Propriedade "Mínimo". Para cada tag é definido um tipo para o seu valor. Os seguintes tipos incorporados estão disponíveis:

- Digital
- AnalogInt
- AnalogDouble
- AnalogDecimal
- Text
- TDateTime
- Counter
- Timer
- Reference

A sintaxe tag.<TagName> está disponível para todos os tipos de tag. Para saber o tipo de um tag em particular é possível acessar propriedades adicionais e métodos usando a sintaxe: Tag.<TagName>.<PropertyName>. O elemento TagObj mostra propriedades e métodos disponíveis para todos os tipos de tag. Por outro lado, Analog mostra as propriedades comuns a todos tags Analógicos (refere-se ao tipo de tag para propriedades específicas). São permitidos os tipos criados pelo usuário em combinação com os tipos incorporados.

ClassTagObj

Classes de base para todos os objetos tag.

ForceValue

Força a propriedade de valor do objeto ao valor dado como parâmetro. Parâmetros: valor do objeto.

Exemplo:

```
@Tag.AnalogInt1.ForceValue(10);     (VB)
@Tag.AnalogInt1.ForceValue(10);     (C#)
```

Tostring

Retorna uma string que representa o objeto atual. Exemplo:

```
Dim s as string;                (VB)
@Tag.AnalogInt1 = 33;
s = Tag.AnalogInt1.Tostring();
string s;                       (C#)
@Tag.AnalogInt1 = 33;
s = Tag.AnalogInt1.Tostring();
```

Neste exemplo, “S” será avaliado como "33".

Alarm Disable

Fornece ou define o estado de alarme desabilitado do Tag. Se 0 (zero), a propriedade AlarmDisable não fica ativa. No caso de valores diferentes de 0 (zero), a propriedade AlarmDisable fica ativa. Quando a propriedade AlarmDisable não está ativa, todos os alarmes associados ao Tag atual não são tratados. Exemplo:

```
@Tag.AnalogDecimal1.AlarmDisable = 1;       (VB)
@Tag.AnalogDecimal1.AlarmDisable = 1;       (C#)
```

Neste exemplo, os alarmes associados ao objeto AnalogDecimal1 não serão tratados.

Alarm State

Fornece ou define o estado de alarme do Tag. Se 0 (zero): fora de estado de alarme. No caso de valores diferentes de 0 (zero): em estado de alarme. Exemplo:

```
Dim alarmState as Int;                       (VB)
alarmState = @Tag.AnalogDecimal1.AlarmState;
int alarmState;                              (C#)
alarmState = @Tag.AnalogDecimal1.AlarmState;
```

GetName

Fornece o nome do Tag. Exemplo:

```
@Tag.ReferenceTagName.Link = @Tag.TagName.GetName();  (VB)
@Tag.ReferenceTagName.Link = @Tag.TagName.GetName();  (C#)
```

AlarmUnack

Fornece ou define o estado de reconhecimento do Tag atual. Se 0 (zero): alarme atual reconhecido. No caso de valores diferentes de 0 (zero): alarme atual não reconhecido. Exemplo:

```
Dim alarmStatus as Int;                        (VB)
alarmStatus = @Tag.AnalogDecimal1.AlarmUnack;
int alarmStatus;                               (C#)
alarmStatus = @Tag.AnalogDecimal1.AlarmUnack;
```

Changed

Fornece ou define o estado alterado do valor do Tag. True: valor alterado. False: valor inalterado.

Domain

Fornece ou define a propriedade de Domínio do Tag. Se 0 (zero): servidor. Se 1: cliente.

EngUnits

Fornece ou define as unidades de engenharia usadas para quantificar o Tag.

Format

Fornece ou define o formato do valor do Tag para fins de exibição.

Historian

Fornece o objeto HistoryItem onde o Tag atual está configurado como TagName. Se tags duplicados forem permitidos nas tabelas de Historiadores, então o Historiador apontará para o último objeto HistoryItem onde o Tag atual está configurado como TagName. Exemplo:

```
@Tag.AnalogInt1.Historian.Deadband = 3;       (VB)
@Tag.AnalogInt1.Historian.Deadband = 3;       (C#)
```

Locked

Fornece ou define o estado bloqueado do Tag. Quando um Tag está bloqueado, o valor usado para o processamento origina-se da propriedade LockValue e não da propriedade de valor. Se 0 (zero): desbloqueado. No caso de valores diferentes de 0 (zero): bloqueado. Exemplo:

```
@Tag.AnalogDouble1.Locked = 1;        (VB)
@Tag.AnalogDouble1.Locked = 1;        (C#)
```

Quality

Fornece ou define o estado de qualidade do Tag. Valores permitidos:

Valor	Qualidade
0	Ruim
192	Bom

Tabela 6-9. Valores Padrões

Exemplo:

```
@Tag.AnalogInt1.Quality = 192;      (VB)
@Tag.AnalogInt1.Quality = 192;      (C#)
```

Retentive

Fornece ou define a propriedade Retentiva do Tag, a qual especifica se a propriedade de Valor do Tag deve ser salva quando a aplicação for encerrada. O valor salvo é então usado como o valor de inicialização na execução da próxima aplicação. Se 0 (zero): não retentivo. No caso de valores diferentes de 0 (zero): retentivo. Exemplo:

```
@Tag.AnalogInt1.Retentive = 1;      (VB)
@Tag.AnalogInt1.Retentive = 1;      (C#)
```

TimeStamp

Fornece ou define a propriedade Timestamp do tag. Exemplo:

```
Dim dt coo DateTimeOffset;           (VB)
dt = @Tag.AnalogDouble1.Timestamp;
DateTimeOffset dt;                   (C#)
dt = @Tag.AnalogDouble1.Timestamp;
```

ValueType

Fornece a propriedade ValueType do Tag. Exemplo:

```
Dim doubleType as Integer;                 (VB)
doubleType = @Tag.AnalogDouble1.ValueType;
int doubleType;                            (C#)
doubleType = @Tag.AnalogDouble1.ValueType;
```

Visibility

Fornece ou define o estado de visibilidade do Tag. Se 0 (zero): privado, 1: protegido e 2: público. Exemplo:

```
Dim visibilityState as Integer;                    (VB)
```

```
visibilityState = @Tag.AnalogDouble3.Visibility;
int visibilityState;                                  (C#)
visibilityState = @Tag.AnalogDouble3.Visibility;
```

Class Digital

Propriedades do Runtime para Classe Digital. Valores possíveis: 0 = false e 1 = true. Equivalente no script:

- C#: int32
- VB: int
- .NET: int

Toogle

Alterna a propriedade de valor do Tag entre 0 (zero) e 1. Se o valor atual é 0 (zero), então o novo valor é 1. Se o valor atual é 1, então o novo valor é 0 (zero). Exemplo:

```
Dim newValue as Byte;                     (VB)
newValue = @Tag.Digital1.ToggleValue();
byte newValue;                            (C#)
newValue = @Tag.Digital1.ToggleValue();
```

Lock

Fornece ou define a propriedade LockedValue do Tag. Quando um Tag está bloqueado, o valor usado para o processamento origina-se da propriedade LockValue e não da propriedade Valor. Exemplo:

```
@Tag.Digital1.LockValue = 1;      (VB)
@Tag.Digital1.LockValue = 1;      (C#)
```

State

Fornece a propriedade de Estado do Tag digital. Valor = 0 corresponde ao estado = False (VB) ou estado = false (C#). Valor = 1 corresponde ao estado = True (VB) ou estado= true (C#).

Value

Fornece ou define a propriedade de Valor do Tag digital. Valores válidos: 0 (zero) ou 1. Exemplo:

```
@Tag.Digital1.Value = 1;     (VB)
@Tag.Digital1.Value = 1;     (C#)
```

Class Analog

Propriedades do Runtime para Analog Class.

Membro	Descrição
Bit0	Fornece ou define o bit 0 do valor do tag.
Bit1	Fornece ou define o bit 1 do valor do tag.
Bit10	Fornece ou define o bit 10 do valor do tag.
Bit11	Fornece ou define o bit 11 do valor do tag.
Bit12	Fornece ou define o bit 12 do valor do tag.
Bit13	Fornece ou define o bit 13 do valor do tag.
Bit14	Fornece ou define o bit 14 do valor do tag.
Bit15	Fornece ou define o bit 15 do valor do tag.
Bit16	Fornece ou define o bit 16 do valor do tag.
Bit17	Fornece ou define o bit 17 do valor do tag.
Bit18	Fornece ou define o bit 18 do valor do tag.
Bit19	Fornece ou define o bit 19 do valor do tag.
Bit2	Fornece ou define o bit 2 do valor do tag.
Bit20	Fornece ou define o bit 20 do valor do tag.
Bit21	Fornece ou define o bit 21 do valor do tag.
Bit22	Fornece ou define o bit 22 do valor do tag.
Bit23	Fornece ou define o bit 23 do valor do tag.
Bit24	Fornece ou define o bit 24 do valor do tag.
Bit25	Fornece ou define o bit 25 do valor do tag.
Bit26	Fornece ou define o bit 26 do valor do tag.
Bit27	Fornece ou define o bit 27 do valor do tag.
Bit28	Fornece ou define o bit 28 do valor do tag.
Bit29	Fornece ou define o bit 29 do valor do tag.
Bit3	Fornece ou define o bit 3 do valor do tag.
Bit30	Fornece ou define o bit 30 do valor do tag.
Bit31	Fornece ou define o bit 31 do valor do tag.
Bit4	Fornece ou define o bit 4 do valor do tag.
Bit5	Fornece ou define o bit 5 do valor do tag.
Bit6	Fornece ou define o bit 6 do valor do tag.
Bit7	Fornece ou define o bit 7 do valor do tag.
Bit8	Fornece ou define o bit 8 do valor do tag.
Bit9	Fornece ou define o bit 9 do valor do tag.
Hi	Fornece ou define o limite high do tag.
HiHi	Fornece ou define o limite high-high do tag.
Lo	Fornece ou define o limite low do tag.
LoLo	Fornece ou define o limite low-low do tag

Tabela 6-10. Propriedades do Runtime para Analog Class

Class Analog<T>

Propriedades de Runtime para Classe Analog <T>.

Deadband

Fornece ou define a banda morta do Tag analógico. Exemplo:

```
@Tag.AnalogDouble1.Deadband = 5;       (VB)
@Tag.AnalogDouble1.Deadband = 5;       (C#)
```

LockValue

Fornece ou define o valor de bloqueio do Tag analógico. Exemplo:

```
@Tag.AnalogDouble1.LockValue = 50;      (VB)
@Tag.AnalogDouble1.LockValue = 50;      (C#)
```

Analog.Min

Fornece ou define o valor máximo do Tag analógico. Exemplo:

```
@Tag.AnalogDouble1.Max = 100;          (VB)
@Tag.AnalogDouble1.Max = 100;          (C#)
```

StartValue

Fornece ou define o valor inicial do tag analógico. Exemplo:

```
@Tag.AnalogDouble1.StartValue = 50;          (VB)
@Tag.AnalogDouble1.StartValue = 50;          (C#)
```

State

Fornece ou define o estado do Tag analógico. Se o valor é igual a 0 (zero), o estado é FALSE. Se o valor é diferente de 0 (zero), o estado é TRUE. Exemplo:

```
Dim state as Boolean;               (VB)
state = @Tag.AnalogDouble1.State;
bool state;
state = @Tag.AnalogDouble1.State;  (C#)
Value
@Tag.AnalogDouble1.Value = 55;     (VB) ou
@Tag.AnalogDouble1 = 55;
@Tag.AnalogDouble1.Value = 55;     (C#) ou
@Tag.AnalogDouble1 = 55;
```

Class AnalogInt

Propriedades do Runtime para AnalogInt Class.

Class AnalogDecimal

Propriedades do Runtime para Classe Decimal Analógica.

Class AnalogDoble

Propriedades do Runtime para Classe AnalogDouble.

Class Text

Propriedades do runtime para classe texto.

LockValue

Fornece ou define o valor de bloqueio do tag de texto. Exemplo:

```
@Tag.Text1.LockValue = "Welcome";         (VB)
@Tag.Text1.LockValue = "Welcome";         (C#)
```

Value

Fornece ou define o valor do tag de texto.

Exemplo:

```
@Tag.Text1.Value = "My text";        (VB) ou
@Tag.Text1 = "My text";
@Tag.Text1.Value = "My text";        (C#) ou
@Tag.Text1 = "My text";
```

Class TDataTime

Propriedades do Runtime para ClasseTDataTime.

LockValue

Fornece ou define o valor de bloqueio do Tag de TDataTime. Exemplo:

```
@Tag.Text1.LockValue = "Welcome";          (VB)
@Tag.Text1.LockValue = "Welcome";          (C#)
```

Value

Fornece ou define o valor do Tag de TDataTime. Exemplo:

```
@Tag.Text1.Value = "My text";        (VB) ou
@Tag.Text1 = "My text";
@Tag.Text1.Value = "My text";        (C#) ou
@Tag.Text1 = "My text";
```

Class Counter

Define propriedades do Runtime para Classe Counter.

Event

Fornece o evento do Tag Counter. Valores possíveis: "Change", "ChangeUp" e "ChangeDown". Exemplo:

```
Dim counter1Event as string;           (VB)
counter1Event = @Tag.Counter1.Event;
string counter1Event;                  (C#)
counter1Event = @Tag.Counter1.Event;
```

Model

Fornece o modelo do Tag Counter. Valores possíveis: "Up" e "Down". Exemplo:

```
Dim counter1Model as string;            (VB)
counter1Model = @Tag.Counter1.Model;
string counter1Model;                   (C#)
counter1Model = @Tag.Counter1.Model;
```

Trigger

Fornece ou define o disparo do Tag Counter. Exemplo:

```
@Tag.Counter1.Trigger = "Tag.Digital1";     (VB)
@Tag.Counter1.Trigger = "Tag.Digital1";     (C#)
```

Class Timer

Propriedades do Runtime para Classe Timer.

Interval

Fornece ou define o intervalo do Tag Timer. Trata-se de uma string que representa o intervalo de tempo exibido no formato "hh:mm:ss.mmm". Exemplo:

```
@Tag.Timer1.Interval = "0:0:10";         (VB)
@Tag.Timer1.Interval = "0:0:10";         (C#)
```

Model

Fornece o modelo do Tag Timer. Valores possíveis: "SquareWave", "Pulse" e "Comparer". Exemplo:

```
Dim timerModel as string;              (VB)
timerModel = @Tag.Timer1.Model;
string timerModel;                     (C#)
timerModel = @Tag.Timer1.Model;
```

Class Reference

Propriedades de Runtime para a Classe Reference.

Link

Fornece ou define o link do Tag Reference. Exemplo:

```
@Tag.Reference1.Link = @Tag.TagName.GetName();     (VB)
@Tag.Reference1.Link = @Tag.TagName.GetName();     (C#)
```

Class TDataTable

Propriedades do Runtime para Classe TDataTable.

Initialize

Define uma nova referência ao objeto Tabela. Este método é usado somente internamente.

Table

Fornece uma cópia do objeto DataTable. Exemplo:

```
Dim dt as New TDataTable(parent, id);           (VB)
Dim table As DataTable;
table = dt.Table;
TDataTable dt = new TDataTable(parent, id);     (C#)
DataTable table;
table = dt.Table;
```

OverwriteOnUpdate

Fornece ou define o operando OverwriteOnUpdate. Esta propriedade é usada somente internamente.

Update

Atualiza o objeto Tabela. Este método é usado somente internamente. Parâmetros: DataTable table

Classe UserType

Propriedades do Runtime para Classe UserType.

Namespace Security

Class ModuleSecurity

Propriedades do Runtime para objetos ModuleSecurity.

GetPasswordHint

Retorna (exibe) a dica de senha para o nome do usuário selecionado. Parâmetro: string userName. Exemplo:

```
Dim pswHint As string = @Security.GetPasswordHint("User"); (VB)
string pswHint = @Security.GetPasswordHint("User");         (C#)
```

AddRuntimeUser

Acrescenta um usuário à lista de usuários de Runtime. Parâmetros associados: string name; string permissionsStr; string password; string passwordHint; string policyStr; string profileEmail; string profilePhone e string profileCompleteName. Exemplo:

```
@Security.AddRuntimeUser("User", "User", "psw", "remember", "Default",
"a@b.com", "67521855", "Alfred Burns");   (VB)
@Security.AddRuntimeUser("User", "User", "psw", "remember", "Default",
"a@b.com", "67521855", "Alfred Burns");  (C#)
```

GetListOfUserNames

Retorna a lista de nomes dos usuários de Runtime separados por \n (nova linha). Exemplo:

```
Dim userList as string;                               (VB)
userList = @Security.GetListOfUserNames();
string userList;                                      (C#)
userList = @Security.GetListOfUserNames();
```

RemoveRuntimeUser

Remove o usuário de Runtime da lista de usuários. Parâmetro: string name. Exemplo:

```
@Security.RemoveRuntimeUser("User");     (VB)
@Security.RemoveRuntimeUser("User");     (C#)
```

Permission

Fornece a lista de permissão. Refere-se aos objetos SecurityPermission. Exemplo:

```
Dim permissionList as SecurityPermissionList;   (VB)
permissionList = @Security.Permission;
SecurityPermissionList permissionList;          (C#)
permissionList = @Security.Permission;
```

User

Fornece a lista de usuários. Fornece acesso aos objetos SecurityUser. Exemplo:

```
Dim userList as SecurityUserList;  (VB)
userList = @Security.User;
SecurityUserList userList;         (C#)
userList = @Security.User;
```

Policy

Fornece a lista de Política. Exemplo:

```
Dim policyList as SecurityPolicyList;     (VB)
policyList = @Security.Policy;
SecurityPolicyList policyList;            (C#)
policyList = @Security.Policy;
```

Class SecurityPermission

Propriedades do Runtime para objetos SecurityPermission.

DateCreated

Fornece a data e hora em que a SecurityPermission foi criada. Exemplo:

```
Dim permissionDate as DateTime;                  (VB)
permissionDate = @Security.Permission.Administrator.DateCreated;
DateTime permissionDate;                         (C#)
permissionDate = @Security.Permission.Administrator.DateCreated;
```

DateModified

Fornece a data e hora em que a SecurityPermission foi modificada. Exemplo:

```
Dim permissionDate as DateTime;          (VB)
permissionDate = @Security.Permission.Administrator.DateModified;
DateTime permissionDate;                 (C#)
permissionDate = @Security.Permission.Administrator.DateModified;
```

Description

Fornece a descrição da SecurityPermission. Exemplo:

```
Dim permissionDescription as string;       (VB)
permissionDescription = @Security.Permission.Administrator.Description;
int permissionDescription;                 (C#)
permissionDescription = @Security.Permission.Administrator.Description;
```

Edit

Fornece a permissão de Edição. Exemplo:

```
Dim editPermission as Integer;        (VB)
editPermission = @Security.Permission.Administrator.Edit;
int editPermission;                   (C#)
editPermission = @Security.Permission.Administrator.Edit;
```

Name

Fornece o nome SecurityPermission. Exemplo:

```
Dim permissionName as string;      (VB)
permissionName = @Security.Permission.Administrator.Name;
string permissionName;             (C#)
permissionName = @Security.Permission.Administrator.Name;
```

Run

Fornece a permissão de execução. Exemplo:

```
Dim runPermission as Integer;       (VB)
```

```
runPermission = @Security.Permission.Administrator.Run;
int runtPermission;                  (C#)
runPermission = @Security.Permission.Administrator.Run;
```

Class SecurityUser

Propriedades do Runtime para objetos SecurityUser.

DateCreated

Fornece a data e hora em que o SecurityUser foi criado. Exemplo:

```
Dim userDate as DateTime;         (VB)
userDate = @Security.User.Guest.DateCreated;
DateTime userDate;                (C#)
permissionDate = @Security.User.Guest.DateCreated;
```

DateModified

Fornece a data e hora em que o SecurityUser foi modificado. Exemplo:

```
Dim userDate as DateTime;       (VB)
userDate = @Security.User.Guest.DateModified;
DateTime userDate;              (C#)
userDate = @Security.User.Guest.DateModified;
```

PolicyName

Fornece o nome das políticas do SecurityUser. Exemplo:

```
Dim userPolicyName as string;        (VB)
userPolicyName = @Security.User.Guest.PolicyName;
string userPolicyName;                (C#)
userPolicyName = @Security.User.Guest.PolicyName;
```

SecurityUser.Blocked

Fornece o estado bloqueado do SecurityUser. Exemplo:

```
Dim blockedState as Boolean;      (VB)
blockedState = @Security.User.Guest.Blocked;
bool blocked;                     (C#)
blocked = @Security.User.Guest.Blocked;
```

SecurityUser.Deleted

Fornece o estado excluído do SecurityUser. Exemplo:

```
Dim deletedState as Boolean;    (VB)
deletedState = @Security.User.Guest.Deleted;
bool deleted;                   (C#)
deleted = @Security.User.Guest.Deleted;
```

SecurityUser.Name

Fornece o nome do SecurityUser. Exemplo:

```
Dim userName as string;     (VB)
userName = @Security.User.Guest.Name;
string userName;            (C#)
userName = @Security.User.Guest.Name;
```

SecurityUser.PasswordHint

Fornece a dica de senha dos SecurityUsers. Exemplo:

```
Dim pswHint as string;      (VB)
pswHint = @Security.User.Guest.PasswordHint;
string pswHint;             (C#)
pswHint = @Security.User.Guest.PasswordHint;
```

SecurityUser.Permissions

Fornece as permissões do SecurityUser. Exemplo:

```
Dim userPermissions as Long;        (VB)
userPermissions = @Security.User.Guest.Permissions;
long userPermissions;                (C#)
userPermissions = @Security.User.Guest.Permissions;
```

SecurityUser.PermissionsName

Fornece o nome das permissões do SecurityUser. Exemplo:

```
Dim permissionsName as string;      (VB)
permissionsName = @Security.User.Guest.Permissions;
string permissionsName;             (C#)
permissionsName = @Security.User.Guest.Permissions;
```

SecurityUser.Policy

Fornece as políticas do SecurityUser. Exemplo:

```
Dim userPolicy as Long;     (VB)
userPolicy = @Security.User.Guest.Policy;
long userPolicy;            (C#)
userPolicy = @Security.User.Guest.Policy;
```

SecurityUser.Profile

Fornece o perfil do SecurityUser. Exemplo:

```
Dim userProfile as string;      (VB)
userProfile = @Security.User.Guest.Profile;
string userProfile;             (C#)
userProfile = @Security.User.Guest.Profile;
```

Namespace Alarm

Objetos de Runtime e métodos relacionados ao módulo de alarme.

Class AlarmGroup

Propriedades do Runtime para objetos de Grupo de Alarmes.

AckAll

Reconhece todos os itens de alarme que pertencem a este grupo. Alterna a propriedade definida para AckAll. Valores permitidos: 0: reconhecimento se o valor era 1e 1: reconhecimento se o valor era 0. Exemplo:

```
@Alarm.AckAll = (@Alarm.AckAll==0) ? 1 : 0;
```

AckRequired

Fornece o Reconhecimento do operador exigido definido. Se 0: sem exigência de reconhecimento. Se 1: reconhecimento exigido. Exemplo:

```
int AckReq = @Alarm.Group.Critical.AckRequired;
se (AckReq == 1) @Alarm.AckAll = (@Alarm.AckAll==0) ? 1 : 0;
```

Colors

Representa as cores de primeiro plano e plano de fundo para cada estado de alarme. Esta propriedade representa o valor configurado na coluna Cores.

Description

Descrição do Grupo de Alarmes configurada em EditAlarmsGroups. Exemplo:

```
@Tag.string = @Alarm.Group.Critical.Description;
```

Disable

Habilita/Desabilita um Grupo de Alarmes. Se 0: habilita um Grupo de Alarmes. Se 1: desabilita um Grupo de Alarmes. Exemplo:

```
@Alarm.Group.Critical.Disable = 1;
```

Id

Fornece a ID de um Grupo de Alarmes. Exemplo:

```
@Tag.Int = @Alarm.Group.Critical.Id;
```

LogEvents

Fornece o tipo de arquivamento de Historiador em eventos de Alarme. Os valores permitidos são:

- 0 - Nenhum
- 1 - Ativo
- 2 - ActiveAck
- 3 - ActiveNorm
- 4 - All

Exemplo:

```
@Tag.Int = @Alarm.Group.Critical.LogEvents;
```

PriorityItem

Retorna o item de Alarme de maior prioridade de um Grupo de Alarmes. Exemplo:

```
@ALARM.PRIORITYITEM PITEM = @ALARM.GROUP.CRITICAL.PRIORITYITEM;
```

Show

Fornece um valor configurado para exibir/ocultar um Alarme. Valores permitidos:

- 0 - Nenhum
- 1 - Lista

Exemplo:

```
@Tag.Int = @Alarm.Group.Critical.Show;
```

Sound

Propriedade para descrever se o Som do Grupo de Alarmes está habilitado ou não. Valores permitidos:

- 0 - Nenhum
- 1 - Beep

Exemplo:

```
@Tag.Int = @Alarm.Group.Critical.Sound;
```

TotalCount

Fornece o número de Alarmes Ativos. Exemplo:

```
@Tag.Int = @Alarm.Group.Critical.TotalCount;
```

UnAckCount

Fornece o número de Alarmes Sem Reconhecimento. Exemplo:

```
@Tag.Int = @Alarm.Group.Critical.UnAckCount;
```

Class AlarmItem

Propriedades do Runtime para objetos de Itens de Alarme.

AckTime

Horário em que o Alarme está definido em "Reconhecimento".

ActiveTime

Horário em que o Item de Alarme iniciou.

Alarm

Propriedade para verificar se o Item de Alarme está ativo.

ColorBG

Configurada a Cor do Plano de Fundo do Item de Alarme.

ColorFG

Configurada a Cor do Primeiro Plano do Item de Alarme.

Condition

Configura a condição de avaliação para gerar um Item de Alarme. Os valores permitidos estão descritos a seguir.

- Hi (alto)
- HiHi (alto-alto)
- Lo (baixo)
- LoLo (baixo-baixo)
- RateOfChange (taxa de variação)
- DeviationMinor (desvio para mais)
- DeviationMajor (desvio para menos)

- Equal (igual)
- GreaterThan (maior que)
- GreaterEqual (maior ou igual)
- LessThan (menor que)
- LessEqual (menor ou igual)
- Changed (alterado)
- ChangedUp (alterado para cima)
- ChangedDown (alterado para baixo)

Deadband

Banda morta de um item de alarme definido. Configura o tempo de inatividade do item de alarme. O valor é escolhido pelo usuário.

Disable

Propriedade para desabilitar/habilitar um item de alarme. Valores permitidos:

- 0 - Habilitar
- 1 - Desabilitar

Group

Um Grupo ao qual pertence um Item de Alarme.

Id

ID do objeto (uso interno).

Limit

Fornece o valor definido para avaliar as condições do item de Alarme.

Message

Fornece a mensagem configurada para ser exibida quando ocorre o Alarme.

NormTime

Tempo que o Alarme leva para voltar ao "Normal".

Priority

Fornece uma Prioridade do Item de Alarme. O valor é escolhido pelo usuário.

Setpoint

Fornece o valor definido para avaliar as condições do Item de Alarme.

SetPointDeadband

Representa a banda morta para a propriedade SetPoint. Usada para as seguintes condições de alarme: DeviationMinor e DeviationMajor.

State

Fornece o estado de um item de alarme. Um item de alarme pode ser: Ativo, Reconhecimento, Normal e TagName. Neste último caso fornece o TagName definido que será avaliado para gerar um Item de Alarme.

UnAck

Propriedade Ler/Definir um item de alarme de reconhecimento. Valores permitidos:

- 0 - Não Reconhecer Item de Alarme
- 1 - Reconhecer Item de Alarme

Class ModuleAlarm

Métodos do Runtime e propriedades para o módulo de alarme.

AckAll

Dispara esta propriedade para reconhecer todos os Alarmes.

BeepState

Indica se o beep está configurado para ON (valor = 1) no computador cliente.

Group

Fornece acesso aos objetos do Grupo de Alarmes.

InitializationMessage

Lê/Define a Mensagem Inicial a ser exibida.

Item

Acesso aos objetos do Item de Alarme.

LastErrorMessage

Contém a mensagem de descrição do erro mais recente ocorrido no Módulo Alarme.

LastStoredTimeStamp

Contém o TimeStamp do erro mais recente ocorrido no Módulo Alarme.

PriorityItem

Referência ao Item de Alarme online de maior prioridade.

QueryActive

Objeto TDataTable com a lista atual de Alarmes ativos.

TotalCount

Contagem total de Alarmes ativos.

UnAckCount

Contagem total de Alarmes que exigem reconhecimento.

Namespace Device

Objetos Runtime e métodos relacionados ao Dispositivo.

Class DeviceAccessType

Propriedades do Runtime para objetos DeviceAccessType.

AcceptUnsolicited

Quando configurado como true, os pontos do dispositivo são autorizados a receber mensagens não solicitadas. O canal do dispositivo deve ter a propriedade AcceptUnsolicited configurada para "true" para permitir mensagens não solicitadas.

ReadOnStartup

Quando configurado como true, o ponto do dispositivo será lido na inicialização do Módulo Dispositivo.

ReadPooling

Quando configurado como true, indica que o Pooling lido está habilitado para este ponto do dispositivo.

ReadPoolingRate

Indica a taxa de Pooling para o grupo de pontos.

WriteEnable

Quando configurado como true, o ponto do dispositivo será escrito para o dispositivo quando o valor deste se alterar.

Class DeviceChannel

Propriedades do Runtime para objetos DeviceChannel.

Activity

Indicação de atividade. O módulo dispositivo alterna este valor para cada operação completada neste Canal.

LastErrorCode

Último Código do Estado de Erro (mais recente) ocorrido neste canal. Valores permitidos:

Valor	Descrição
0	Success
-1	BuildCommandException
-2	ParseCommandUnsolicitedException
-3	ParseReplyException
-4	BuildReplyUnsolicitedException
-5	ChannelException
-6	NodeException
-100	Base Send Error
-101	Base SendAndWait Error
-102	TCP Create Error 1
-103	TCP Create Error 2
-104	TCP Create SocketError
-105	TCP Connect Callback Error
-106	TCP Receive Error
-107	UDP Create Error
-108	UDP Receive Error
-109	Serial Create Error
-110	Serial Receive Error
-111	TCP NotConnected
-112	Start message timeout
-113	Receiving bytes timeout
-114	End message timeout
-115	Connect timeout
-200	ProtocolError
-201	InvalidProtocol
-202	InvalidStation
-203	InvalidCommand
-204	InvalidMsgSequence
-205	InvalidCheckSum
-206	InvalidAddress
-207	InvalidModifiers

Tabela 6-11. Códigos de Estado para Canal ou Nó

Nota:

Valor: Valores positivos são códigos de erro de protocolos específicos.

LastErrorDateTime

TimeStamp do último erro (mais recente) ocorrido neste canal.

Status

Status atual para este canal. Valores permitidos podem ser vistos na Tabela 6-11.

Class DeviceNode

Propriedades do Runtime para objetos DeviceNode.

Activity

Indicação de atividade. O dispositivo alterna este valor para cada operação executada neste Nó.

BackupStation

Estação de backup atual para este Nó.

IsBackup

Indicação de estação de Backup ativa.

IsPrimary

Indicação de PrimaryStation ativa.

IsRedundancyEnabled

Indicação de redundância de Nó.

LastErrorCode

Último estado de código de erro (mais recente) ocorrido neste Nó. Valores permitidos podem ser vistos na Tabela 6-11.

LastErrorDateTime

DateTime para o último (mais recente) erro neste Nó.

PrimaryStation

Estação principal atual para este Nó.

Status

Status atual para este Nó. Valores permitidos podem ser vistos na Tabela 6-11.

Class ModuleDevice

Métodos do Runtime e propriedades para o Módulo Dispositivo.

AccessType

Acesso aos objetos DeviceAccessType.

Channel

Acesso aos objetos DeviceChannel.

Node

Acesso aos objetos DeviceNode.

Namespace Dataset

Objetos de Runtime e métodos relacionados à base de dados.

Class DatasetDB

Propriedades do Runtime para objetos DatasetDB.

Connectionstring

String usada para conectar-se com o banco de dados.

Database

Nome do banco de dados do objeto DB.

Id

Identificação (ID) do objeto (Uso Interno).

LogonName

Nome de logon usada para conectar-se com o banco de dados.

Provider

Provider do banco de dados selecionado.

Class DatasetFile

Propriedades do Runtime para objetos DatasetFile.

LoadCommand

Carrega os valores dos tags configurados na propriedade Objetos a partir do arquivo indicado pela propriedade Nome do arquivo. O parâmetro associado é string statusMessage, ou seja, a mensagem com o status do comando de carga.

SaveCommand

Salva os valores dos tags configurados na propriedade Objetos a partir do arquivo indicado pela propriedade Nome do Arquivo. O parâmetro associado é string statusMessage ou seja, a mensagem com o status do comando de salvamento.

Completed

O valor desta propriedade é aumentado quando uma operação é concluída.

Description

Fornece a descrição do DatasetFile configurado.

Disable

Desabilita os comandos para o DatasetFile quando o valor é maior que zero. Valores permitidos:

- 0 = Habilita os comandos para o DatasetFile
- 1 = Desabilita os comandos para o DatasetFile

FileName

Caminho completo do arquivo que será criado ou carregado. Esta propriedade representa o valor configurado na coluna FileName.

FileType

Indica o formato do arquivo. Valores permitidos:

- ASCII = 0
- Unicode = 1
- XML = 2

Esta propriedade representa o valor configurado na coluna FileType.

Id

Identificação (ID) do objeto (Uso Interno).

LastStatus

Fornece o status da última (mais recente) operação assíncrona. Valores permitidos:

- Zero = Êxito
- Diferente de zero = Código de erro

LastStatusMessage

Fornece a mensagem de status da última (mais recente) operação assíncrona.

Load

Envia um comando Carregar assíncrono quando o valor é alterado. O valor da propriedade LoadExecuted é alterado quando a operação é concluída. Exemplo:

```
if (@Dataset.File.File1Unicode.Load.Equals(0))
    @Dataset.File.File1Unicode.Load = 1;
else
    @Dataset.File.File1Unicode.Load = 0;
```

LoadExecuted

O valor desta propriedade é alterado quando o comando Carregar assíncrono se completa.

Objects

Contém os tags e índices a serem salvos ou carregados. Os valores permitidos são TagName, o qual representa o nome do Tag (o índice inicial também pode ser especificado) e Index, indicando o índice (se o Tag é um array, e o índice inicial é especificado). Exemplo:

TagName	Índice
Tag.doubleArray[0]	10
Tag.textArray[2]	5
Tag.SelectStatusMsg	

Tabela 6-12. Exemplo da Opção Objects

Save

Envia um comando Salvar quando o valor é alterado. O valor da propriedade SaveExecuted é alterado quando a operação se conclui. Exemplo:

```
if (@Dataset.File.File1Unicode.Save.Equals(0))
    @Dataset.File.File1Unicode.Save= 1;
else
    @Dataset.File.File1Unicode.Save= 0;
```

SaveExecuted

O valor desta propriedade é alterado quando o comando de salvamento assíncrono se completa.

Class DatasetQuery

Propriedades do Runtime para objetos DatasetQuery.

ExecuteCommand

Executa um comando síncrono de acordo com o SqlStatement. Os valores de retorno em relação ao status da operação estão indicados a seguir:

- Zero = Êxito
- Diferente de zero = Código do erro

ExecuteCommandWithStatus

Executa um comando síncrono de acordo com o SqlStatement; exibe uma mensagem de status. Os valores de retorno em relação ao status da operação estão indicados a seguir:

- Zero = Êxito
- Diferente de zero = Código do erro

Parâmetros: out string statusMessage. A mensagem indica o status do comando Next.

NextCommand

Executa um comando síncrono Next que aumenta o valor da propriedade CursorIndex. Os valores de retorno em relação ao status da operação estão indicados a seguir:

- Zero = Êxito
- Diferente de zero = Código do erro

Os tags configurados na coluna de mapeamento receberão o valor da próxima linha.

NextCommandWithStatus

Executa um comando síncrono Next que aumenta o valor da propriedade CusorIndex e exibe uma mensagem de status. Os valores de retorno em relação ao status da operação estão indicados a seguir:

- Zero = Êxito
- Diferente de zero = Código do erro

O parâmetro associado é out string statusMessage, ou seja, a mensagem que indica o status do comando Next. Os tags configurados na coluna mapeamento receberão o valor da próxima linha.

SelectCommand

Executa um comando síncrono Selecionar, de acordo com o SqlStatement. Os valores de retorno são: DataTable em caso de êxito; caso contrário, nulo. Os tags configurados na coluna mapeamento receberão o valor da primeira linha. Exemplo:

```
DataTable dataTable = @Dataset.Query.query1.SelectCommand();
if (dataTable != null && dataTable.Rows.Count > 0)
{
    @Tag.firstItem = dataTable.Rows[0]["Item"].Tostring();
}
```

SelectCommandWithStatus

Executa um comando síncrono Selecionar, de acordo com o SqlStatement, e exibe informação de status. Os valores de retorno são: DataTable em caso de êxito; caso contrário, nulo. O primeiro parâmetro associado é: out int status. O status da operação está indicado a seguir:

- Zero = Êxito
- Diferente de zero = Código do erro

O segundo parâmetro associado é: out string statusMessage, ou seja, a mensagem que indica o status do comando Select. Os tags configurados na coluna mapeamento receberão o valor da primeira linha. Exemplo:

```
int status;
string statusMessage;
DataTable dataTable;
    dataTable = @Dataset.Query.query1.SelectCommandWithStatus(out status,
    out statusMessage);
    if (status == 0 && dataTable != null && dataTable.Rows.Count > 0))
    {
        @Tag.firstItem = dataTable.Rows[0]["Item"].Tostring();
    }
```

AsyncContents

Contém a TDataTable resultante de um dos comandos assíncronos, tais como Select ou Update.

Completed

O valor desta propriedade aumenta quando uma operação assíncrona se conclui.

CursorIndex

Definir a posição da linha atual no DatasetTable resultante.

DB

Fornece o DB configurado em EditDatasetsTables.

Description

Fornece a descrição do DatasetQuery.

Disable

Desabilita os comandos para o DatasetQuery quando o valor é maior que zero. Valores permitidos:

- 0 = Habilita os comandos para o DatasetQuery
- 1 = Desabilita os comandos para o DatasetQuery

Execute

O valor da propriedade Execute se altera quando a operação é concluída. Exemplo:

```
if (@Dataset.Query.Query1.Execute.Equals(0))
    @Dataset.Query.Query1.Execute = 1;
else
    @Dataset.Query.Query1.Execute = 0;
```

ExecuteCompleted

O valor da propriedade ExecuteCompleted se altera quando a operação é concluída.

Id

Identificação (ID) do Objeto (Uso Interno).

LastStatus

Fornece o status da última (mais recente) operação assíncrona. Valores permitidos:

- Zero = Êxito
- Diferente de zero = Código do erro

LastStatusMessage

Fornece a mensagem de status para a última (mais recente) operação assíncrona, onde uma string vazia indica êxito.

Mapping

Fornece o mapeamento das colunas do DataTable resultantes com os tags. Valores de retorno:

TagName	Coluna
int_Id	ID
txt_Name	Nome
txt_Description	Descrição

Tabela 6-13. Mapeamento das Colunas do DataTable

Resultado:

```
Tag.int_Id=ID;
Tag.txt_Name=Name;
Tag.txt_Description=Description;
```

Next

Envia um comando Next assíncrono quando o valor é alterado. O valor da propriedade NextExecuted se altera quando a operação é concluída. Exemplo:

```
if (@Dataset.Query.Query1.Next.Equals(0))
```

```
    @Dataset.Query.Query1.Next = 1;
else
    @Dataset.Query.Query1.Next = 0;
```

NextExecuted

O valor desta propriedade se altera quando o comando assíncrono Next se completa.

RowCount

Fornece o número total de linhas na DatasetTable resultante.

Select

Envia um comando Select assíncrono quando o valor é alterado. O valor da propriedade SelectExecuted se altera quando a operação é concluída. Exemplo:

```
if (@Dataset.Query.Query1.Select.Equals(0))
    @Dataset.Query.Query1.Select= 1;
else
    @Dataset.Query.Query1.Select= 0;
```

SelectExecuted

O valor desta propriedade se altera quando o comando Select assíncrono se completa. O valor da propriedade SelectExecuted se altera quando a operação é concluída.

SqlStatement

Define o comando SQL a ser executado.

Class DatasetTable

Propriedades do Runtime para objetos DatasetTable.

DeleteCommand

Exclui a linha atual da DatasetTable. Os valores de retorno em relação ao status da operação estão indicados a seguir:

- Zero = Êxito
- Diferente de zero = Código do erro

O índice da linha atual é definido pela propriedade CursorIndex.

DeleteCommandWithStatus

Exclui a linha atual da DatasetTable e fornece uma mensagem de status. Os valores de retorno em relação ao status da operação estão indicados a seguir:

- Zero = Êxito
- Diferente de zero = Código do erro

Parâmetros: out string statusMessage. A mensagem indica o status do comando de exclusão onde uma string vazia significa êxito. O índice da linha atual é definido pela propriedade CursorIndex.

InsertCommand

Insere os valores dos tags configurados na coluna mapeamento na DatasetTable na posição indicada pela propriedade CursorIndex. Os valores de retorno em relação ao status da operação estão indicados a seguir:

- Zero = Êxito
- Diferente de zero = Código do erro

InsertCommandWithStatus

Insere os valores dos tags configurados na coluna mapeamento na DatasetTable na posição indicada pela propriedade CursorIndex e fornece uma mensagem de status. Os valores de retorno em relação ao status da operação estão indicados a seguir:

- Zero = Êxito
- Diferente de zero = Código do erro

O parâmetro associado é: out string statusMessage, ou seja, a mensagem que indica o status do comando de inserção, onde uma string vazia significa êxito.

NextCommand

Executa um comando Next síncrono o qual aumenta o valor da propriedade CursorIndex. Os valores de retorno em relação ao status da operação estão indicados a seguir:

- Zero = Êxito
- Diferente de zero = Código do erro

Os tags configurados na coluna mapeamento receberão o valor da próxima linha.

NextCommandWithStatus

Executa um comando Next síncrono o qual aumenta o valor da propriedade CursorIndex e fornece uma mensagem de status. Os valores de retorno em relação ao status da operação estão indicados a seguir:

- Zero = Êxito
- Diferente de zero = Código do erro

O parâmetro associado é: out string statusMessage, ou seja, a mensagem que indica o status do comando Next, onde uma string vazia significa êxito. Os tags configurados na coluna mapeamento receberão o valor da próxima linha.

SelectCommand

Executa um comando Select síncrono na DatasetTable. Valor de retorno: DataTable em caso de êxito; caso contrário é nulo. Os tags configurados na coluna mapeamento receberão o valor da primeira linha. Exemplo:

```
DataTable dataTable = @Dataset.Table.table1.SelectCommand();

    if (dataTable != null && dataTable.Rows.Count > 0)
    {
        @Tag.firstItem = dataTable.Rows[0]["Item"].Tostring();
    }
```

SelectCommandWithStatus

Executa um comando Select síncrono na DatasetTable e fornece uma mensagem de status. Valor de retorno: DataTable em caso de êxito; caso contrário é nulo.

Parâmetro 1: out int status. Status da operação de seleção:

- Zero = sucesso
- Diferente de zero = código de erro

Parâmetros 2: out string statusMessage. A mensagem indica o status do comando Selecionar, onde uma string vazia significa sucesso. Os tags configurados na coluna mapeamento receberão o valor da primeira linha. Exemplo:

```
int status;
string statusMessage;
DataTable dataTable;
dataTable = @Dataset.Table.table1.SelectCommandWithStatus(out status, out
statusMessage);

    if(status == 0 && dataTable != null && dataTable.Rows.Count > 0))
    {
        @Tag.firstItem = dataTable.Rows[0]["Item"].Tostring();
    }
```

UpdateCommand

Atualiza a linha atual da DatasetTable com os valores dos tags configurados na coluna mapeamento. Os valores de retorno são: DataTable em caso de êxito; caso contrário é nulo. O índice da linha atual é definido pela propriedade CursorIndex.

UpdateCommandWithStatus

Atualiza a linha atual da DatasetTable com os valores dos tags configurados na coluna mapeamento e fornece uma mensagem de status. Valor de retorno: DataTable em caso de êxito; caso contrário é nulo.

Parâmetro 1: out int status. Status da operação de atualização:

- Zero = êxito
- Diferente de zero = código de erro

Parâmetro 2: out string statusMessage. A mensagem indica o status do comando de atualização, onde uma string vazia significa êxito. O índice da linha atual é definido pela propriedade CursorIndex.

Access

Tipo de Acesso ao DatasetTable. Valores permitidos:

- 0 - Ler
- 1 - Inserir
- 2 - Ler/Escrever
- 3 - Irrestrito

AsyncContents

Contém a TDataTable resultante de um dos comandos assíncronos, tais como <i>Select ou Update.

Completed

O valor desta propriedade aumenta quando uma operação assíncrona se conclui.

CursorIndex

Define a posição da linha atual na DatasetTable.

DB

Fornece o DB configurado em EditDatasetsTables.

Delete

Envia um comando Delete assíncrono quando o valor se altera.

O valor da propriedade DeleteExecuted é alterado quando a operação é concluída. Exemplo:

```
if (@Dataset.Table.table1.Delete.Equals(0))
    @Dataset.Table.table1.Delete= 1;
else
    @Dataset.Table.table1.Delete = 0;
```

DeleteExecuted

O valor desta propriedade se altera quando o comando Delete assíncrono se completa.

Description

Fornece a Descrição do DatasetTable.

Disable

Desabilita os comandos para a DatasetTable quando o valor é maior que zero. Valores permitidos:

- 0 = Habilita os comandos para o DatasetTable
- 1 = Desabilita os comandos para o DatasetTable

Id

Identificação (ID) do Objeto (Uso Interno).

Insert

Envia um comando Insert assíncrono quando o valor se altera. O valor da propriedade InsertExecuted altera quando a operação é concluída. Exemplo:

```
if (@Dataset.Table.table1.Insert.Equals(0))
    @Dataset.Table.table1.Insert= 1;
else
    @Dataset.Table.table1.Insert = 0;
```

InsertExecuted

O valor desta propriedade é alterado quando o comando Insert assíncrono se completa.

LastStatus

Fornece o status da última (mais recente) operação assíncrona. Valores permitidos:

- Zero = êxito
- Diferente de zero = código de erro

LastStatusMessage

Fornece a mensagem de status da última (mais recente) operação assíncrona, onde uma string vazia significa êxito.

Mapping

Fornece o mapeamento das colunas da DataTable com os tags. Exemplo:

TagName	Coluna
int_Id	ID
txt_Nome	Nome
txt_Descrição	Descrição

Tabela 6-14. Mapping

Resultado:

```
Tag.int_Id=ID;
Tag.txt_Name=Name;
Tag.txt_Description=Description;
```

Next

Envia um comando Next assíncrono quando o valor é alterado. O valor da propriedade NextExecuted altera quando a operação é concluída. Exemplo:

```
if (@Dataset.Table.table1.Next.Equals(0))
   @Dataset.Table.table1.Next= 1;
else
   @Dataset.Table.table1.Next = 0;
```

NextExecuted

O valor desta propriedade é alterado quando o comando assíncrono Next se completa.

RowCount

Fornece o número total de linhas na DatasetTable.

Select

Envia um comando Select assíncrono quando o valor é alterado.

SelectExecuted

O valor da propriedade SelectExecuted se altera quando a operação é concluída. Exemplo:

```
if (@Dataset.Table.table1.Select.Equals(0))
   @Dataset.Table.table1.Select = 1;
else
   @Dataset.Table.table1.Select = 0;
```

TableName

Fornece o nome da tabela de dados.

Update

Envia um comando Update assíncrono quando o valor é alterado. O valor da propriedade UpdateExecuted altera quando a operação é concluída. Exemplo:

```
if (@Dataset.Table.table1.Update.Equals(0))
   @Dataset.Table.table1.Update = 1;
```

```
else
   @Dataset.Table.table1.Update = 0;
```

UpdateExecuted

O valor desta propriedade é alterado quando o comando Update assíncrono se completa.

WhereCondition

Define a condição Onde para a DatasetTable. A cláusula SQL WHERE é usada para selecionar dados condicionalmente. Exemplo:

Tipo	Preço($)	DateAdded
1	27	10/02/2008
1	120	07/10/2010
2	50	12/01/2009

Tabela 6-15. SQL WHERE

```
@Dataset.Table.table1.WhereCondition = "DateAdded > '01/01/2010'";
```

O comando retornará somente os itens onde DateAdded for maior que 01/01/2010

Class ModuleDataset

Métodos do Runtime e propriedades para o Módulo Dataset.

DB

Fornece acesso aos objetos DatasetDB.

File

Fornece acesso aos objetos DatasetFile.

Query

Fornece acesso aos objetos DatasetQuery.

Table

Fornece acesso aos objetos DatasetTable.

Namespace Script

Objetos Runtime e métodos relacionados à programação (Script).

Class ModuleScript

Métodos do Runtime e propriedades para o Módulo Script.

RunTasksSimultaneous

Habilita a execução simultânea de tarefas (Multi-Threading). Valores permitidos:

- 0 - Execução simultânea NÃO permitida
- 1 - Execução simultânea permitida

Task

Acesso aos objetos ScriptTask.

Sintaxe:

```
Script.Task.<TaskName>.
```

UserClass

Acesso aos objetos ScriptUserClass.

Sintaxe:

```
Script.UserClass.<UserClassName>.
```

Class ScriptTask

Classe de Tarefas Script.

Description

Fornece a Descrição da ScriptTask configurada em EditScriptsTasks.

Disable

Desabilita a execução da ScriptTask quando o valor é maior que zero. Valores permitidos:

- Zero = êxito
- Diferente de zero = código de erro

Domain

Indica o Domínio da ScriptTask. Os valores permitidos são 0: Servidor e 1: Cliente. No caso de Servidor o ScriptTask é executado no contexto do Servidor e não pode acessar os objetos Cliente, isto é, telas ou tags de domínio do cliente. Já no caso de Cliente o ScriptTask é executado em cada Cliente onde o ModuleScript esteja em execução.

ErrorMessage

Contém a mensagem com o último (mais recente) erro ocorrido na execução do script.

Event

Reservado para uso futuro.

ExecutionsCount

Fornece a quantidade de execuções desde que o módulo foi iniciado.

LastCPUTime

Fornece o tempo da UCP utilizado durante a última (mais recente) execução do script.

LastExecution

Contém o TimeSpan da última (mais recente) execução do script.

Period

Período de tempo exigido para executar uma Tarefa. Esta propriedade representa o valor configurado em EditScriptsTasks.

Running

Indica se o script atual está em execução. Valores permitidos:

- 0 - ScriptTask NÃO está em execução
- 1 - ScriptTask está em execução

StopExecutionOnError

Indica se a execução do script deve ser interrompida em caso de erro.

Trigger

Fornece o Tag ou Objeto que dispara a execução da Tarefa. Esta propriedade representa o valor configurado em EditScriptsTasks.

Namespace Display

Objetos Runtime e métodos relacionados às telas da aplicação.

Class Display

Propriedades do Runtime para objetos Tela. Sintaxe: Display.<DisplayName>

Close

Fecha a tela selecionada.

Open

Abre a tela selecionada.

Description

Fornece a descrição da tela configurada em EditDisplaysDisplays.

Id

Identificação (ID) do Objeto (Uso Interno).

IsOpened

Indica se a tela selecionada está aberta.

Class Layout

Propriedades do Runtime para objetos Layout. Sintaxe: Layout.<LayoutName>.

OpenCommand

Abre o layout selecionado.

Description

Fornece a descrição do layout, configurado em EditDisplaysLayouts.

Id

ID do Objeto (Uso Interno).

IsOpened

Indica se o layout selecionado está aberto.

Namespace Report

Objetos de Runtime e métodos relacionados ao Módulo Relatório.

Class ModuleReport

Acesso aos objetos <ReportItem>.

Class ReportItem

Propriedades de Runtime para objetos Relatório.

OpenCommand

Abre o relatório selecionado. Valores de retorno:

- True = êxito
- False = erro

O ReportViewer pode ser usado para visualizar o conteúdo do relatório.

SaveCommand

Salva o relatório selecionado no caminho indicado pela propriedade SaveFileName. Valores de retorno:

- True = êxito
- False = erro

Append

Indica se o relatório será sobrescrito ou acrescentado nos comandos Salvar. Valores permitidos:

- 0 - Relatório sobrescrito nos comandos Salvar
- 1 - Relatório acrescentado nos comandos Salvar

Completed

O valor desta propriedade aumenta quando a operação Salvar ou Carregar é concluída.

Description

Fornece a descrição do ReportItem configurada em EditReportsReports.

Disable

Desabilita as operações ReportItem quando o valor é maior que zero. Valores de retorno:

- Zero = Habilita o ReportItem
- Maior que zero = Desabilita o ReportItem

Id

Identificação (ID) do Objeto (Uso Interno).

LastStatus

Fornece ou define o status do último ReportItem processado. Valores permitidos:

- Success = 0
- InvalidMode = 1
- Disabled = 2
- NoObjectsConfigured = 3
- ReportException = 4
- InvalidContent = 5
- ModuleStopped = 6
- ModulePaused = 7

LastStatusMessage

Fornece ou define a mensagem de status do último ReportItem processado. Valores permitidos:

- "Success"
- "InvalidMode"
- "Disabled"
- "NoObjectsConfigured"
- "ReportException"
- "InvalidContent"
- "ModuleStopped"
- "ModulePaused"

OpenExecuted

Fornece ou define o estado OpenExecuted no relatório. Valores permitidos:

- 0 (zero) - Comando Abrir não executado
- 1 - Comando Abrir executado

Padding

Fornece o valor de ajuste no Relatório. Valores permitidos:

- Compact = 0
- PadRight = 1
- PadLeft = 2

SaveExecuted

Fornece ou define o estado SaveExecuted no relatório. Valores permitidos:

- 0 (zero) - Comando Salvar não executado
- 1 - Comando Salvar executado

SaveFileName

Fornece ou define o caminho completo usado ao salvar o Relatório.

SaveFormat

Fornece ou define o formato de salvamento no Relatório. Valores permitidos:

- XPS = 0
- Html = 1
- Unicode = 2
- ASCII = 3
- PDF = 4

Namespace Info

Objetos Runtime e métodos relacionados às informações do projeto.

Class ModuleInfo

Trace

Rastreia um objeto do sistema. As referências a este objeto são exibidas na janela de rastreamento. Parâmetro: string text. Exemplo:

```
@Info.Trace("Digital1")            (VB)
@Info.Trace("Digital1");           (C#)
```

License

Fornece o objeto InfoLicense que representa a licença atual.

ExecutionPath

Fornece o caminho de execução onde estão instalados os módulos executáveis do sistema.

Module

Fornece o objeto InfoModuleList que representa a lista dos módulos.

OnlineConfig

Fornece o estado de configuração online. Valores permitidos:

- TRUE - configuração online habilitada
- FALSE - configuração online desabilitada

Project

Fornece o objeto InfoProjectVersion que representa a informação da versão do projeto.

ProjectSettings

Fornece o objeto InfoProjectSettings que representa a informação das configurações do projeto.

TestMode

Fornece o estado do modo teste do projeto. Valores permitidos:

- TRUE - Projeto está no modo teste
- FALSE - Projeto não está no modo teste

TestModeSync

Fornece o estado de sincronização do modo teste. Se 0 (zero), a sincronização do modo teste está desabilitada. Se diferente de 0 (zero), a sincronização do modo teste está habilitada. A sincronização do modo Teste permite ao Modo Teste trabalhar com os valores disponíveis dos tags atuais a partir da execução da inicialização.

Nota:

TestModeSync: Esta funcionalidade está disponível apenas na versão Enterprise.

Class InfoprojectVersion

CurrentBuild

Fornece o número atual do compilador.

DateCreated

Fornece o objeto DateTime que representa a data em que a versão do projeto foi criada.

DateModified

Fornece o objeto DateTime que representa a data em que a versão do projeto foi modificada.

Description

Fornece a descrição da versão do projeto.

ProductFamily

Fornece a família do produto desta versão do projeto. Valores possíveis:

- "Student"
- "Express"
- "Lite"
- "Enterprise"

ProductModel

Fornece o modelo do produto da versão do projeto. Valores possíveis:

- 75
- 150
- 300
- 500
- 1500
- 2500
- 5000
- 15000
- 25000
- 50000
- 100000
- 1000000

ProductName

Fornece o nome do produto da versão do projeto. Valor possível: "BluePlant".

ProductVersion

Fornece a versão do produto desta versão do projeto.

ProjectName

Fornece o nome do projeto desta versão do produto.

ProjectPath

Fornece o caminho desta versão do projeto.

TargetFramework

Fornece o framework destino desta versão do projeto.

Versionstring

Fornece a string desta versão do projeto. Exemplo: "1.0 (Editing)".

Class InfoProjectSettings

CultureInfo

Fornece o idioma do projeto.

Class InfoModuleList

Alarm

Fornece o objeto InfoModule para o módulo Alarme.

DataSet

Fornece o objeto InfoModule para o módulo DataSet.

Device

Fornece o objeto InfoModule para o módulo Dispositivo.

Display

Fornece o objeto InfoModule para o módulo Tela.

Historian

Fornece o objeto InfoModule para o módulo Historiador.

ModuleInformation

Fornece o objeto InfoModule para o módulo ModuleInformation.

OPCServer

Fornece o objeto InfoModule para o módulo OPCServer.

PropertyWatch

Fornece o objeto InfoModule para o módulo PropertyWatch.

Report

Fornece o objeto InfoModule para o módulo Report.

ReportServer

Fornece o objeto InfoModule para o módulo ReportServer.

Script

Fornece o objeto InfoModule para o módulo Script.

Security

Fornece o objeto InfoModule para o módulo Segurança.

Server

Fornece o objeto InfoModule para o módulo Servidor.

TraceWindow

Fornece o objeto InfoModule para o módulo TraceWindow.

Class InfoLicense

AllowedRichClients

Fornece o estado AllowedRichDevices. Valores permitidos:

- Zero - clientes Rich não permitidos
- Diferente de zero - clientes Rich permitidos

AllowedRunInstances

Fornece o estado AllowedRunInstances. Valores permitidos:

- Zero - instâncias de execução não permitidas
- Diferente de zero - instâncias de execução permitidas

AllowedTagElements

Fornece o estado AllowedTagElements. Valores permitidos:

- Zero - elementos do Tag não permitidos
- Diferente de zero - elementos do Tag permitidos

AllowedWebClients

Fornece o estado AllowedWebClients. Valores permitidos:

- Zero - clientes Web não permitidos
- Diferente de zero - clientes Web permitidos

DateCreated

Fornece a data e hora em que a licença foi criada.

DateModified

Fornece a data e hora em que a licença foi modificada.

ExpirationDate

Fornece a data e hora em que a licença expira.

LicenseMedia

Fornece informação sobre a mídia da licença. Valores possíveis:

- "Hardkey"
- "Softkey"

LicenseType

Fornece informação sobre o tipo da licença. Valores possíveis:

- "None"
- "Run"
- "Eng_Run"
- "Developer"

ProductFamily

Fornece a família do produto da licença. Valores possíveis:

- "Student"
- "Express"
- "Lite"
- "Enterprise"

ProductModel

Fornece um valor numérico que caracteriza o modelo do produto licenciado considerando as seguintes opções: 75, 150, 300, 500, 1.500, 2.500, 5.000, 15.000, 25000, 50.000, 100.000, 1.000.000.

ProductVersion

Fornece a versão de produto da licença.

SerialNumber

Fornece o número de série da licença.

ServerConnected

Fornece o endereço completo do servidor.

AllowedEngineeringUsers

Fornece o estado AllowedEngineeringUsers. Valores permitidos:

- Zero - usuários de engenharia não permitidos
- Diferente de zero - usuários de engenharia permitidos

AllowedDevices

Fornece o estado AllowedDevices. Valores permitidos:

- Zero - dispositivos não permitidos
- Diferente de zero - dispositivos permitidos

Class InfoModule

IsPaused

Fornece ou define o estado IsPaused do módulo. Valores permitidos:

- TRUE - módulo está pausado
- FALSE - módulo não está pausado

IsRunning

Fornece ou define o estado IsRunning do módulo. Valores permitidos:

- TRUE - módulo em execução
- FALSE - módulo não está em execução

Namespace Server

Objetos Runtime e métodos relacionados ao servidor.

ServerStation

A classe ServerStation contém informações sobre o computador no qual o Servidor do Runtime (programa TServer.exe) está sendo executado.

LoadProjectVersion

Carrega o projeto mostrado pelo caminho especificado. O caminho deve apontar para um projeto válido no servidor. Valores permitidos: TRUE em caso de êxito e FALSE em caso de falha. Parâmetro associado: string projectPathName.

SwitchToStandby

Transfere a execução para o computador reserva se a redundância está habilitada. Valores permitidos: TRUE em caso de êxito e FALSE em caso de falha.

HttpAddress

Fornece o endereço http do servidor.

IsPrimary

Fornece o estado IsPrimary. Valores permitidos:

- TRUE - servidor é principal
- FALSE - servidor não é principal

IsRedundancyEnabled

Fornece o estado IsPrimary do servidor. Valores permitidos:

- TRUE - redundância habilitada
- FALSE - redundância desabilitada

IsSecondary

Fornece o estado IsSecondary. Valores permitidos:

- TRUE - servidor é secundário
- FALSE - servidor não é secundário

IsStandByActive

Fornece o estado IsStandByActive. Valor padrão:

- TRUE - servidor reserva está ativo
- FALSE - servidor reserva está inativo

IsSwitchToPrimaryEnabled

Fornece o estado IsSwitchToPrimayEnabled. Valores permitidos:

- TRUE - opção de troca para o principal habilitada
- FALSE - opção de troca para o principal desabilitada

ServerStation.ComputerIP

Fornece ou define o IP do computador do servidor.

ServerStation.ComputerName

Fornece ou define o nome do computador do servidor.

ServerStation.Date

Fornece a data do servidor.

ServerStation.Day

Fornece o dia do mês do servidor.

ServerStation.DayOfWeek

Fornece o dia da semana do servidor.

ServerStation.DayOfYear

Fornece o dia do ano do servidor. Valores permitidos: 1 a 366.

ServerStation.Hour

Fornece a hora do dia do servidor.

ServerStation.Minute

Fornece o componente minuto da data do servidor.

ServerStation.Month

Fornece o componente mês da data do servidor.

ServerStation.Now

Fornece a data local do servidor e offset de tempo.

ServerStation.Second

Fornece o componente segundo da data do servidor.

ServerStation.ShutDown

Fornece ou define o estado de desligamento do servidor. Valores permitidos:

- TRUE - servidor está sendo desligado
- FALSE - servidor não está sendo desligado

ServerStation.Startup

Fornece o estado de inicialização do servidor. Valores permitidos:

- TRUE - servidor foi inicializado
- FALSE - servidor não foi inicializado

ServerStation.Ticks

Fornece o número de marcações que representam a data e hora do servidor.

ServerStation.Time

Fornece a hora do dia do servidor.

ServerStation.Year

Fornece o componente ano da data do servidor.

TimeMs

Fornece a hora do dia do servidor (inclusive milissegundos).

Namespace Client

Objetos de Runtime e métodos relacionados ao Namespace do Cliente.

Class ClientStation

A classe ClientStation contém informações sobre o computador no qual o cliente está em execução (TVisualizer.Exe ou clientes Web).

ChangeUserPassword

Altera a senha do usuário referenciado. Valor de retorno: TRUE em caso de sucesso e FALSE em caso de falha. Parâmetros associados: string username, string oldPassword e string newPassword.

CloseDisplay

Fecha a tela referenciada. Parâmetro associado: string displayName.

GetPasswordHint

Fornece a dica de senha para o usuário referenciado. Valor de Retorno: a dica da senha. Parâmetro associado: string userName.

Locale

Retorna o texto localizado. Valor de Retorno: texto localizado. Parâmetro associado: string text.

LogOn

Executa o logon do usuário especificado com a senha especificada. Parâmetros associados: string username e string password. Valores de retorno:

- OK = 0
- ServerNotAvailable = 1
- InvalidLogon = 2
- ServerNotConnected = 3
- UserBlocked = 4
- UserDeleted = 5
- CannotStartModule = 6
- InvalidUserName = 10
- InvalidPassword = 11
- PermissionsRestrictions = 12
- UnknownError = 99

LogOnGuest

Executa o logon do usuário como "Guest".

OpenDisplay

Abre a tela na página do último layout. Parâmetro associado: string displayName

OpenDisplayAtIndex

Abre a tela na página do layout especificado pelo índice. Parâmetros: string displayName e int index.

OpenLayout

Abre o layout referenciado. O parâmetro associado é: string layoutName.

OpenPopupNote

Abre uma nota de popup com título e descrição especificados. Os parâmetros associados são: string title, string description, bool isReadOnly, double left e double top.

OpenPreviousPage

Abre a página exibida anteriormente. Valores permitidos:

- TRUE em caso de sucesso
- FALSE em caso de falha

SaveLayoutAsImage

Salva o layout como uma imagem. O caminho para o arquivo será definido no diálogo seguinte.

SaveLayoutAsImageFile

Salva o layout como arquivo de imagem. O parâmetro associado é: string fileName.

SwitchToStandby

Troca o servidor para o modo Standby. Os valores permitidos são:

- TRUE em caso de sucesso
- FALSE em caso de falha

BeepOff

Fornece ou define o estado de beep desligado do cliente. Os valores permitidos são:

- TRUE - beep desligado
- FALSE - beep ligado

BlinkFast

Fornece a propriedade de piscar rápido do cliente. Esta propriedade é uma variável digital que alterna de 0 a 1 e de 1 a 0, permanecendo 500 milissegundos em cada estado.

BlinkSlow

Fornece a propriedade de piscar devagar. Esta propriedade é uma varável digital que alterna entre 0 e 1, permanecendo 250 ms em cada estado.

ComputerIP

Fornece o IP do computador.

ComputerName

Fornece o nome do computador.

CultureInfo

Fornece ou define a seleção de idioma do cliente.

CurrentPage

Fornece o nome da página do cliente exibida atualmente.

CurrentUser

Fornece o usuário atual do cliente.

Date

Fornece a data do cliente.

DateTime

Fornece a data e hora do cliente.

Day

Fornece o dia do mês do cliente.

DayOfWeek

Fornece o dia da semana do cliente.

DayOfYear

Fornece o dia do ano do cliente.

Dictionary

Fornece ou define o dicionário do cliente.

Hour

Fornece o componente hora da data do cliente.

InputPassword

Fornece ou define a senha de entrada do cliente. Esta é uma variável auxiliar usada na janela de logon padrão do sistema.

InputUserName

Fornece ou define o nome do usuário de entrada do cliente. Esta é uma variável auxiliar usada na janela de logon padrão do sistema.

IsWebBrowser

Fornece o estado IsWebBrowser do cliente.

LayoutName

Fornece ou define o nome do layout do cliente.

Millisecond

Fornece o componente milissegundo da data do cliente.

Minute

Fornece o componente minuto da data do cliente.

Month

Fornece o componente mês da data do cliente.

Now

Fornece a data local do cliente e offset de tempo.

OnScreenKeyboard

Fornece ou define o estado de tela e teclado do cliente.

PreviousPage

Fornece o nome da página anterior do cliente.

Second

Fornece o componente segundo da data do cliente.

SelectedPage

Fornece ou define o nome da página selecionada do cliente.

ServerHttpAddress

Fornece o endereço http do cliente.

ShutDown

Fornece ou define o estado de desligamento do cliente. Os valores permitidos são:

- TRUE - cliente está desligado
- FALSE - cliente não está desligado

SimulationAnalog

Fornece uma variável de simulação [int] analógica, a qual varia de 0 a 100 (em incrementos de 1); retorna a 0 em um ciclo e então repete o mesmo padrão (onda dente de serra).

SimulationDigital

Fornece uma variável de simulação digital que alterna entre 0 e 1, permanecendo 3 segundos em cada estado.

SimulationDouble

Fornece uma variável de simulação [double] analógica, a qual varia de 0 a 100; retorna de 100 a 0, e então repete o mesmo padrão.

Startup

Fornece o estado de inicialização do cliente. Valores permitidos:

- TRUE - cliente foi inicializado
- FALSE - cliente não foi inicializado

Ticks

Fornece o número de marcações que representam a data e hora do cliente.

Time

Fornece o horário do dia do cliente.

TimeMs

Fornece o horário do dia do cliente incluindo milissegundos.

Tomorrow

Fornece o componente dia da data de amanhã do cliente.

UserName

Fornece o nome de usuário do cliente.

UtcNow

Fornece a data UTC e offset de horário do cliente.

Year

Fornece o componente ano da data do cliente.

Yesterday

Fornece o componente dia da data de ontem do cliente.

CurrentPage

Fornece o nome da pagina atual.

DateTime

Fornece a data e hora da página do cliente exibida atualmente.

IsWebBrowser

Fornece o estado IsWebBrowser do cliente. Os valores permitidos são:

- TRUE - o cliente está em execução com um navegador da Web
- FALSE - o cliente não está em execução com um navegador da Web

OnScreenKeyboard

Fornece ou define o estado de tela e teclado do cliente. Parâmetro associado: public bool OnScreenKeyboard { get; set; }. Os valores permitidos são:

- TRUE - funcionalidade do teclado e tela está ativa
- FALSE - funcionalidade do teclado e tela não está ativa

PreviousPage

Fornece o nome da página anterior do cliente.

ServerHttpAddress

Fornece o endereço http do cliente.

SimulationAnalog

Fornece uma variável de simulação [int] analógica, a qual varia de 0 a 100 (em incrementos de 1); retorna a 0 em um ciclo e então repete o mesmo padrão (onda dente de serra).

SimulationDouble

Fornece uma variável de simulação [double] analógica, a qual varia de 0 a 100; retorna de 100 a 0, e então repete o mesmo padrão.

Configurações Avançadas

Esta seção contém informação adicional sobre os aplicativos do BluePlant incluindo:

- Linha de Comando
- Executando o BluePlant como um Serviço do Windows
- Clientes Remotos
- Instalando o Web Server no IIS

Linhas de Comando

As informações sobre as ferramentas BluePlant e executáveis que podem ser chamados usando linhas de comando e parâmetros específicos estão descritos a seguir.

TStartup

Inicializa o projeto BluePlant.

TVisualizer

Inicializa o Módulo BluePlant Visualizer.

TraceWindow

Inicializa a ferramenta BluePlant TraceWindow.

PropertyWatch

Inicializa a ferramenta de diagnóstico PropertyWatch.

ModuleInformation

Inicializa a ferramenta de diagnóstico Módulo Informação.

DisableTaskSwitchProtection

O usuário precisa executar o arquivo de batch file para instalar o driver do dispositivo que pode bloquear CTRL+ALT+DEL enquanto o TVisualizer está em execução.

RegServer

Registra o servidor BluePlant OPC.

UnRegServer

Cancela o registro do servidor BluePlant OPC.

Executando o BluePlant como um Serviço do Windows

Este procedimento descreve como instalar aplicações BluePlant do servidor Runtime para execução como um serviço do Windows.

Por favor, observe que, para permitir a Engenharia Distribuída e também para servir páginas para clientes Web, o usuário também precisa ter um servidor Web habilitado. Consulte o tópico Installing Web Server in IIS para informações sobre esse procedimento.

A fim de executar o aplicativo como um serviço do Windows:

```
<.NET Framework Install Path>\installutil <Install Path>\<BluePlant
Version>\TStartupAsService.exe.
```

No prompt do DOS (deve-se Executar como Administrador), execute o seguinte comando:

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319>installutil "C:\Program
Files\Altus\bp-2012.1\TStartupAsService.exe"
```

Em seguida, o usuário precisa incluir a linha de comando para o serviço, que é a mesma do programa tStartup.exe: /project:<projectNameAndPath>.

Não está disponível ainda um utilitário para configurar o registro. O usuário precisa fazer isso manualmente. Deve-se definir no Registro do Windows e configurar os parâmetros. Abra o Editor de Registro (regedit) e vá para a seguinte chave:

```
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\TStartup\ImagePath"
"C:\Program Files\Altus\bp-2012.1\TStartupAsService.exe"
"/project:C:\BluePlant Projects\Project1.tproj"
```

Note que nos exemplos acima, deve-se alterar o caminho de instalação do aplicativo para a instalação no seu computador.

Finalmente, nas janelas "Services" (Ferramentas Administrativas), deve-se configurar o "TStartup Service".

Pode-se configurá-lo como "Automatic", de forma que o projeto selecionado abra quando o computador é inicializado.

Clientes Remotos

O único pré-requisito para executar o aplicativo no cliente remoto é a instalação do .Net Framework. O aplicativo não precisa ser instalado nos computadores clientes. Uma vez que o servidor BluePlant tenha o WebServer (IIS ou TWebServer) em execução, pode-se digitar o seguinte endereço do navegador Internet Explorer:

Cliente Rich:

```
http://<server ip or name>/<BluePlant
Version>/tvisualizerremote.application
```

Cliente Web:

```
http://<server ip or name>/<BluePlant Version>/tvisualizerweb.xbap
```

Dependendo da versão do Internet Explorer a seguinte configuração deve ser feita:

Desativar o modo protegido na Zona Internet ou Intranet conforme ilustrado na Figura 6-83.

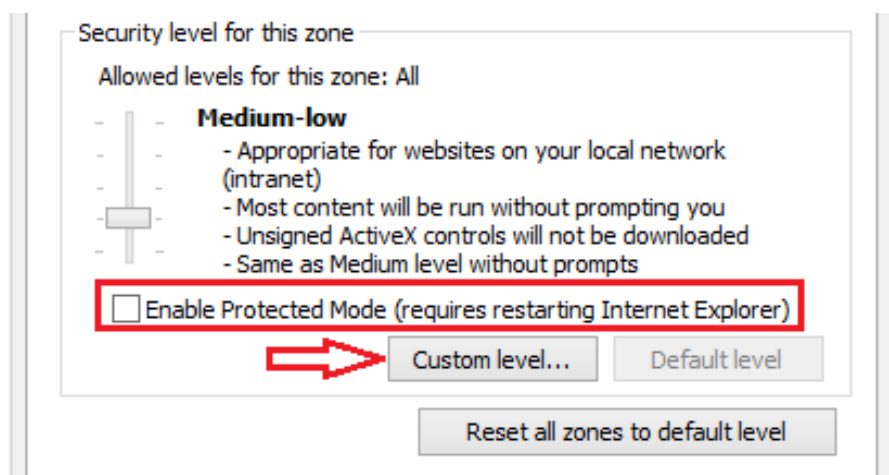



Figura 6-83. Desativar o Modo Protegido

Habilitar aplicações do navegador XAML conforme mostrado na Figura 6-84.

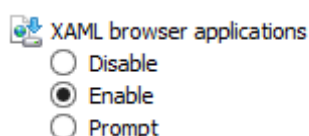



Figura 6-84. Habilitar Aplicações do Navegador XAML

Exemplo:

Cliente Remoto Rich:

http://127.0.0.1/bp-2012.1/TVisualizerRemote.application

Cliente Web:

http://127.0.0.1/bp-2012.1/TVisualizerWeb.xbap

Instalando o Web Server no IIS

Ao instalar o aplicativo, se o IIS for localizado, o usuário não será capaz de instalar o nosso servidor Web embutido (TWebServer.exe).

O programa irá rodar sem o servidor Web, acessando remotamente configurações do projeto e servindo páginas Web em Runtime. Todas as outras ferramentas de execução ou engenharia do BluePlant serão executadas sem a necessidade de instalar o TWebServer ou configurar o IIS.

Se o usuário desejar habilitar o acesso de engenharia remoto e os clientes Web usando o IIS, ele precisa instalar alguns serviços no IIS. Este manual explica como instalar serviços BluePlant no IIS. Esta descrição refere-se ao IIS 7.x, mas outras versões não apresentam muitas diferenças.

Outro cenário em que o IIS é necessário é quando o usuário quer executar o RUNTIME como um serviço do Windows, e ainda ser capaz de fornecer páginas da Web, clientes iPad e engenharia remota.

Pode-se instalar três serviços dentro do IIS, o primeiro deles é o TProjectServer que permite acessar remotamente suas configurações de projeto. O outro serviço é o TVisualizerWeb/TVisualizerRemote que permite o acesso remoto a interfaces em Runtime.. Finalmente, o iDataPanel é o serviço para fornecer dados para dispositivos iOS. Use as seguintes instruções passo-a-passo para configurar o IIS

Procedimento de Instalação

Este procedimento pode ser executado no Windows 7, x64, com o IIS 7.5. Com versões mais antigas do sistema operacional e do IIS, o procedimento pode ser ligeiramente diferente, como a definição de um “Virtual Directory” em vez de “Add Application”.

- Verifique se o IIS está instalado e funcionando, por exemplo, tentando abrir páginas HTML
- Habilite o ASP.Net e manipuladores .Svc para o Serviço WCF

Instalação no IIS 7.x

Se o IIS foi instalado após a instalação do. NET Framework é necessário executar o procedimento descrito em http://msdn.microsoft.com/en-us/library/ms752252.aspx que consiste em executar os seguintes programas usando o prompt de comando como administrador (conforme ilustrado nas figuras a seguir):

`"aspnet_regiis -i -enable" (do diretório de instalação do .NET Framework*)`

```
Administrator: C:\Windows\System32\cmd.exe

C:\Windows\Microsoft.NET\Framework64\v4.0.30319>aspnet_regiis.exe -i -enable
Start installing ASP.NET (4.0.30319).
......
Finished installing ASP.NET (4.0.30319).
```

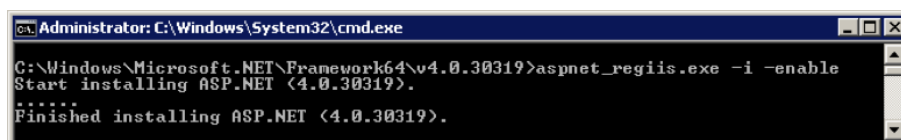



Figura 6-85. Procedimento de Instalação para o IIS 7.x

`"ServiceModelReg.exe" -r (do diretório de instalação do .NET Framework*)`

```
Administrator: C:\Windows\System32\cmd.exe

C:\Windows\Microsoft.NET\Framework64\v4.0.30319>ServiceModelReg.exe -r
Microsoft (R) WCF/WF registration tool version 4.0.0.0
Copyright (c) Microsoft Corporation.  All rights reserved.

Administration utility that manages the installation and uninstallation of
WCF and WF components on a single machine.

[SC] ChangeServiceConfig2 SUCCESS
[SC] ChangeServiceConfig2 SUCCESS
[SC] ChangeServiceConfig2 SUCCESS
[SC] ChangeServiceConfig2 SUCCESS
[SC] ChangeServiceConfig2 SUCCESS
[SC] ChangeServiceConfig2 SUCCESS
[SC] ChangeServiceConfig2 SUCCESS
[SC] ChangeServiceConfig2 SUCCESS
[Warning]The HTTP namespace reservation already exists.

C:\Windows\Microsoft.NET\Framework64\v4.0.30319>_
```

Figura 6-86. ServiceModelReg.exe

O diretório de instalação do .NET Framework é:
C:\Windows\Microsoft.NET\Framework\v4.0.30319, ou
C:\Windows\Microsoft.NET\Framework64\v4.0.30319 (para Windows x64)

Instalação no IIS 8.0

No caso do IIS 8.0 o procedimento está descrito a seguir.

Vá para "Program and Features", escolha “Turn Windows features on or off“.

Figura 6-87. Chaveamento de Funcionalidades do Windows

Habilite a opção "Internet Information Services - World Wide Web Services - Application Development Features - ASP.NET 4.5".

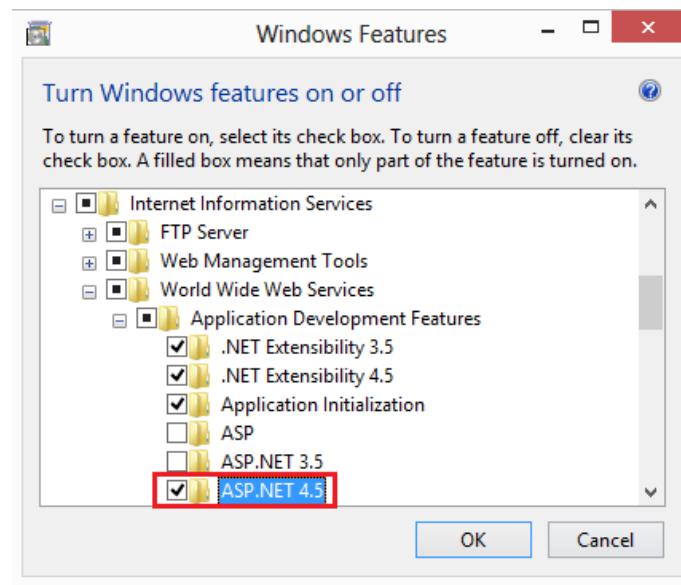



Figura 6-88. Funcionalidades do Windows

Habilite a opção "HTTP Activation" (no .Net Framework 4.5 Advanced Services).

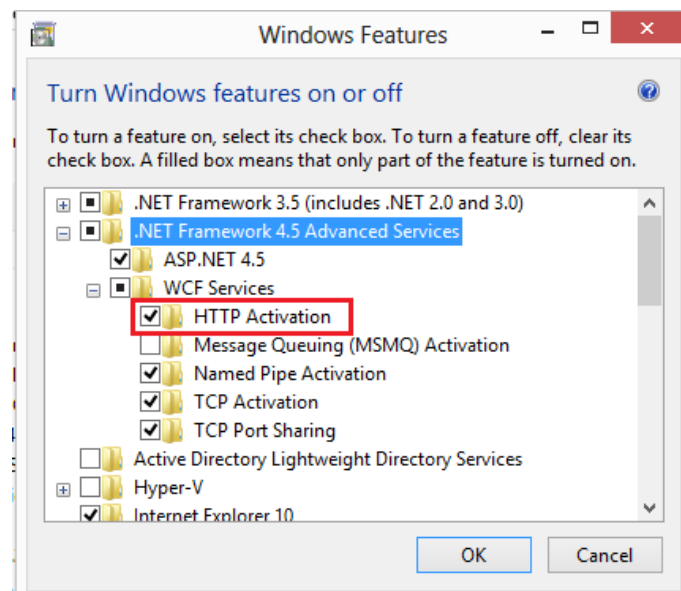



Figura 6-89. Ativação HTTP

Verifique se a extensão ".svc" mapeada para a pasta está mapeada para "aspnet_isapi.dll" (no site msdn.microsoft, link anterior, é explicado como verificar isso para várias versões IIS).

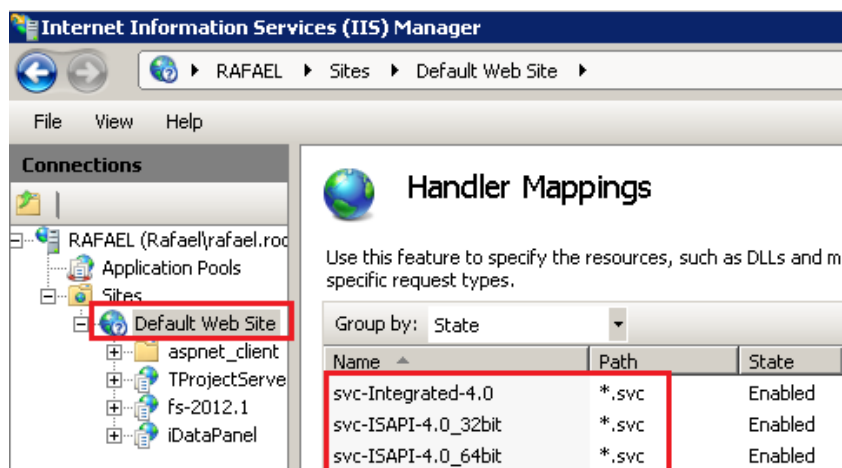



Figura 6-90. Mapeamento da Extensão

Na pasta de instalação do BluePlant, execute o utilitário: InstallTWebServer.exe /uninstall, o qual removerá a instalação atual do TWebServer conforme ilustrado na Figura 6-91.

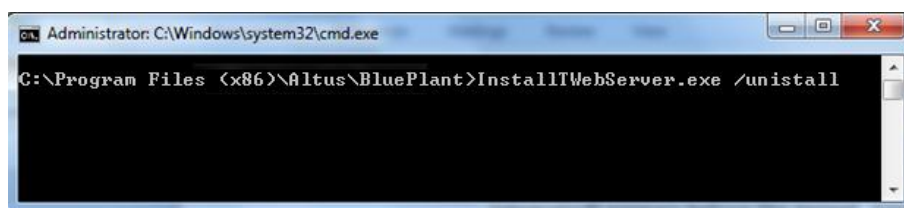



Figura 6-91. Utilitário InstallTWebServer

Na sequência verifique se o TWebServer está definido para iniciar automaticamente. Abra o Editor do Registro do Windows (regedit.exe) e vá para a seguinte chave:

`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run.`

Se for identificada uma referência a "TWebServer", apague-a.

No IIS 7.x, abra "Sites/Default Web Site/Add Application" e configure a informação para os serviços conforme ilustrado na Figura 6-92.

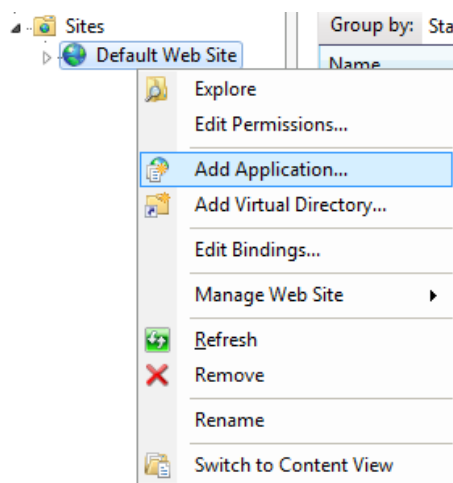



Figura 6-92. Informação para serviços

Os ajustes para o TProjectServer apresentadas na Figura 6-93 são:

- Alias:TProjectServer
- Physical Path (1): verifique e corrija para a pasta de instalação BluePlant
- Application Pool: configure para qualquer conjunto baseado no .NET 4.0

Exemplo: ASP.NET v4.0.

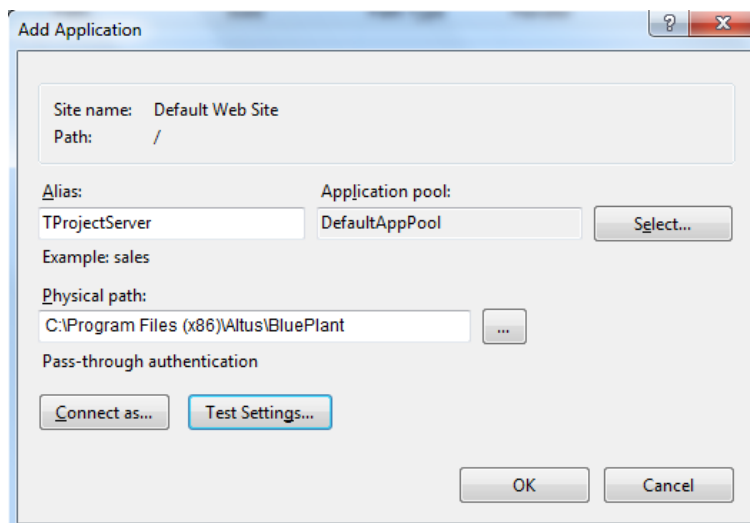



Figura 6-93. ProjectServer

O acesso como "Everyone" (pelo menos para leitura) deve ser configurado. No IE, use a URL "http://localhost/tprojectserver/service.svc" para verificar se o serviço foi corretamente instalado. Ela mostrará uma página com informações sobre o serviço.

Os ajustes para o TVisualizerWeb/TVisualizerRemote são:

- Alias: versão do BluePlant, por exemplo: bp-2012.1
- Physical Path: pasta de instalação do BluePlant para a versão específica
- Application Pool: configure para qualquer conjunto baseado no .NET 4.0
- Exemplo: ASP.NET v4.0

Nota:

Security: A opção Security deve estar habilitada para "Everyone", pelo menos para leitura! No IE, use a URL "http://localhost/bp-2012.1/service.svc" para verificar se o serviço foi corretamente instalado. Ela deve mostrar uma página com informações sobre o serviço.

Os ajustes para o Serviço iDataPanel apresentados Figura 6-94 são:

- Alias: iDataPanel
- Physical Path (1): pasta de instalação do BluePlant para a versão específica
- Application Pool: configure para qualquer conjunto baseado no .NET 4.0

Exemplo: ASP.NET v4.0.

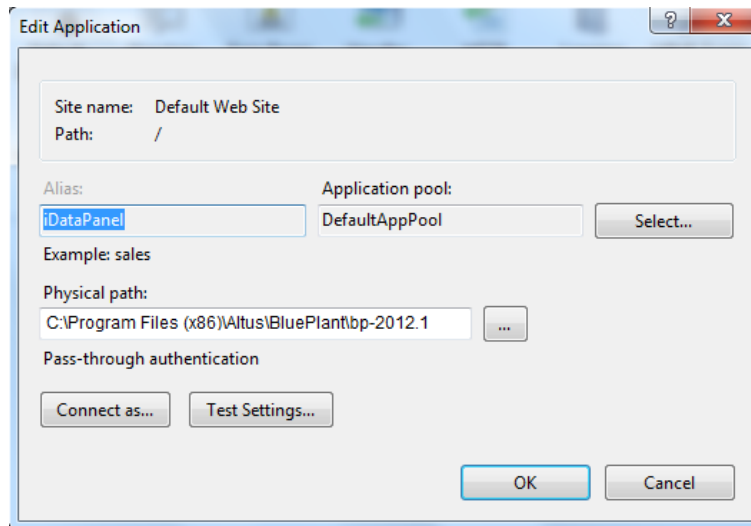



Figura 6-94. Serviço iDataPanel

Nota:

Security: A opção Security deve estar habilitada para "Everyone", pelo menos para leitura. No IE, use a URL "http://localhost/iDataPanel/iDataPanelService.svc" para verificar se o serviço foi corretamente instalado. Ela mostrará uma página com informações sobre o serviço.

Os ajustes para o iDataPanelImages (Diretório Virtual) ilustrados nas figuras a seguir são:

- Alias: iDataPanelImages
- Physical Path: C:\BluePlant Projects\iDataPanelImages

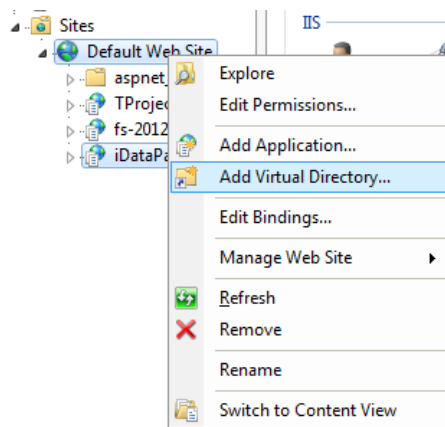



Figura 6-95. Diretório Virtual

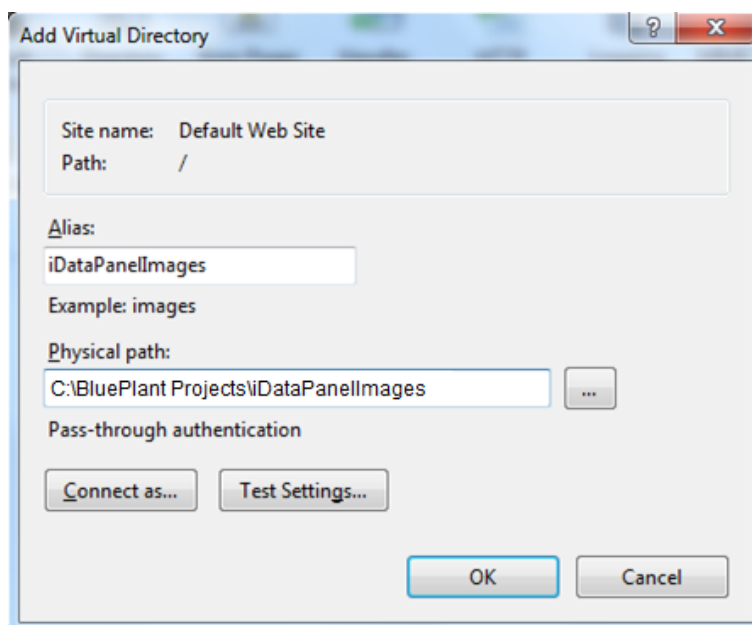



Figura 6-96. Acrescentar Diretório Virtual

Notas:

Security: A opção Security deve estar habilitada para "Everyone", pelo menos para leitura.

“Anonymous Authentication” deve estar habilitada para os serviços conforme pode ser visto na Figura 6-97.

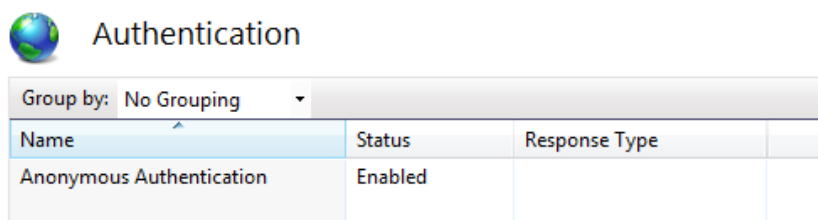



Figura 6-97. Autenticação

O IIS deve ser reiniciado após a configuração dos serviços.

7. Cenários de Sistemas Típicos

Este capítulo descreve os cenários típicos de utilização do BluePlant para atender demandas que contemplam as áreas de produção, utilidades e manufatura.

Além disso, o projeto completo da aplicação pode incluir uma combinação desses cenários com o objetivo de atender as demandas de customização do cliente.

Independentemente do cenário considerado deve-se ter em mente que o BluePlant se constitui em um pacote único de forma que o servidor é sempre um BluePlant. No entanto, todos os módulos do BluePlant como Alarme, Historiador, Dispositivo, Banco de Dados etc, podem estar em computadores diferentes sendo gerenciados pelo Servidor BluePlant gerando assim um sistema distribuído.

Com base nessas premissas podemos ter as configurações descritas a seguir.

Configurações dos Sistemas

Sistema Standalone

Esse sistema caracteriza-se por uma instalação do BluePlant executando o servidor e o cliente SCADA no mesmo computador.

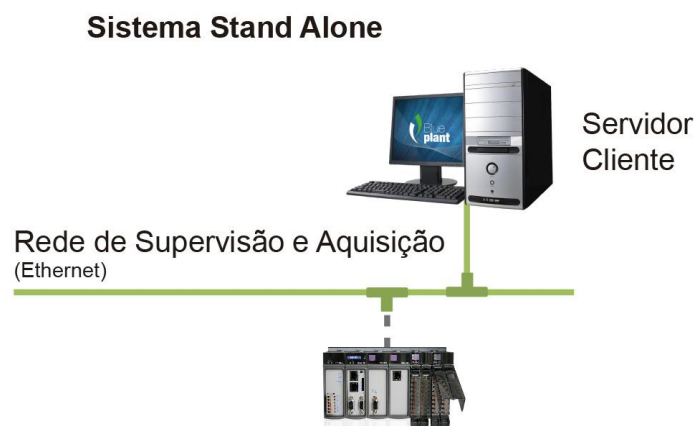



Figura 7-1. Sistema Stand Alone

Sistema de Entradas e Saídas Distribuídas

Esse sistema caracteriza-se por uma máquina BluePlant servidora e módulos de dispositivos rodando em computadores dedicados à comunicação com o processo. Nesse caso o cliente SCADA pode estar no mesmo computador do servidor ou em outro computador. A Figura 7-2 ilustra essa configuração.

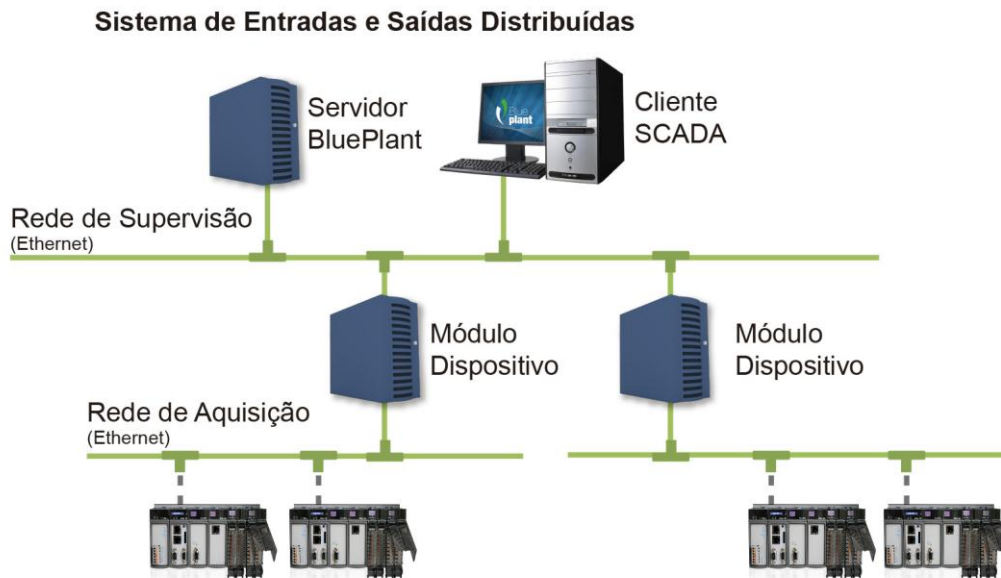



Figura 7-2. Sistema de Entradas e Saídas Distribuídas

Este modelo também é útil em plantas que contêm dispositivos com porta serial ou comunicações com capacidades limitadas. Colocando servidores de E/S no chão de fábrica de interface com esses dispositivos, é possível otimizar as comunicações em redes lentas ou de baixa largura de banda melhorando o desempenho global.

Apesar da distribuição geográfica dos servidores de E/S em várias plantas, este tipo de arquitetura pode ser configurada como um sistema de cluster único, desde que este seja capaz de suportar vários servidores de E/S.

Sistema Cliente e Servidor

Esse sistema contempla um servidor BluePlant onde os módulos Alarme, Historiador, Banco de Dados estão sendo executados e os Clientes SCADA em outros computadores da rede LAN. A Figura 7-3 mostra esse sistema.

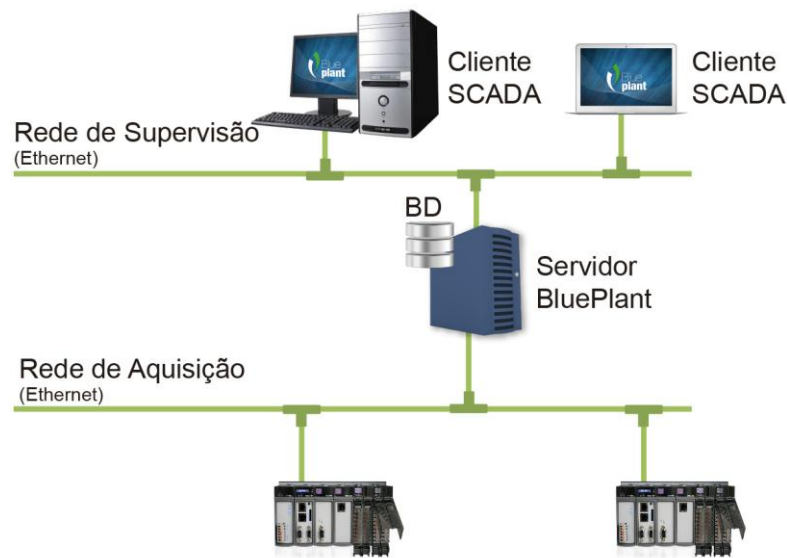



Figura 7-3. Sistema Cliente e Servidor

A arquitetura cliente-servidor permite que os clientes possam ser distribuídos em vários computadores em uma LAN, criando um sistema que oferece benefícios de flexibilidade e desempenho. Este tipo de arquitetura também pode ser configurada como um sistema de cluster único.

Sistema de Servidor Redundante

Neste caso temos dois computadores distintos executando os servidores BluePlant, onde a redundância é feita automaticamente pelo próprio supervisório.. Desta forma é necessário apenas determinar os endereços IP das estações primária e secundária. É possível termos as seguintes configurações de redundância:

- O banco de dados de Alarme e/ou Historiador sendo executado em uma terceira máquina dedicada a históricos, conforme a Figura 7-4.
- Banco de Dados Local nos servidores primário e secundário são utilizados para armazenar os históricos dos módulos de Alarme e/ou Historiador, realizando o sincronismo de dados entre eles de forma automática conforme Figura 7-5.

Figura 7-4. Sistema Servidor Redundante Terceira Máquina

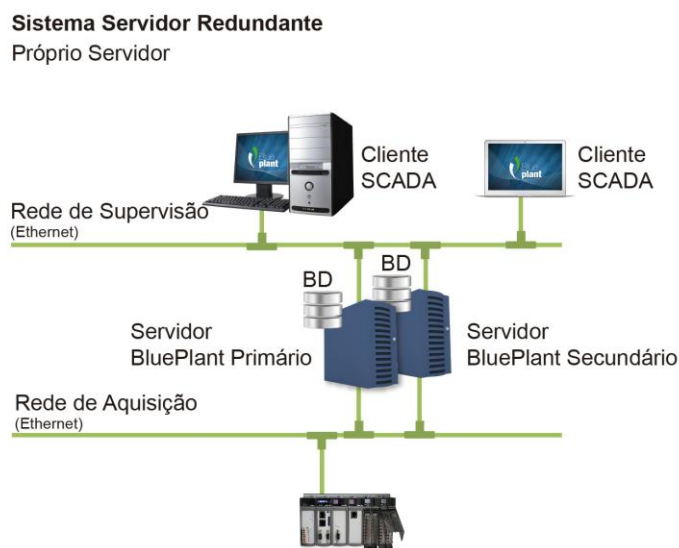



Figura 7-5. Sistema Servidor Redundante Banco de Dados Local

A possibilidade de definir os servidores primário e secundário dentro de um projeto permite a integração do hardware de redundância na infraestrutura do sistema. Isso ajuda a evitar situações onde ocorrerá um erro em um servidor no sistema global, tornando-o inoperante. Sistemas deste tipo são interessantes para garantir operação ininterrupta e confiabilidade na coleta de dados.

Outra configuração possivel é a redundância nos dispositivos de comunicação, onde o(s) servidor(es) tem opcões de comunicar com o dispositivo primário ou com um dispositivo reserva. Essa redundância é habilitada configurando os endereços IPs dos dispositivos. O BluePlant fica responsável por gerenciar a troca entre o dispositivo primário e reserva quando houver algum falha na comunicação entre dispositivo e supervisório.

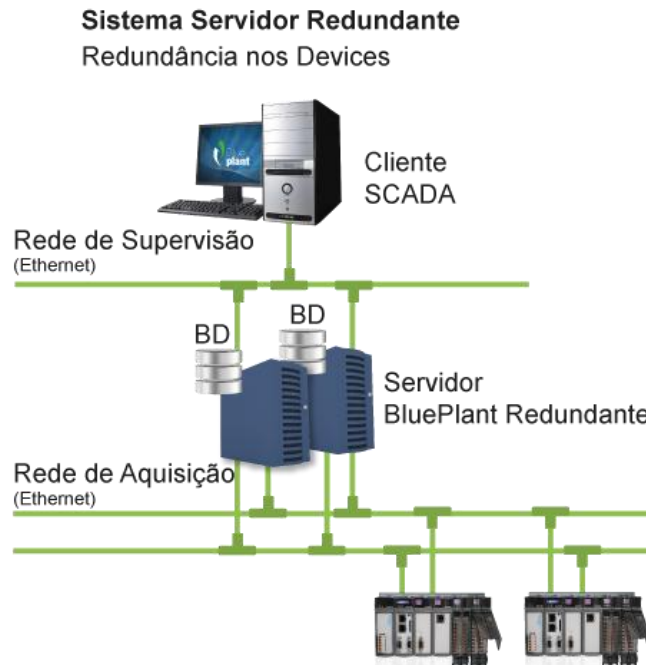



Figura 7-6. Sistema Servidor Redundante com Redundância nos Dispositivos

Sistema de Controle

Neste caso podemos ter vários servidores em plantas distintas (projetos diferentes) em uma arquitetura distribuída. Essa configuração permite que, em uma sala de controle, se possa acessar qualquer uma dessas plantas através de clientes específicos. É importante mencionar que os clientes das plantas não estarão integrados em uma só máquina, de forma que é necessário especificar qual planta deseja-se visualizar. A Figura 7-7 ilustra esse sistema.

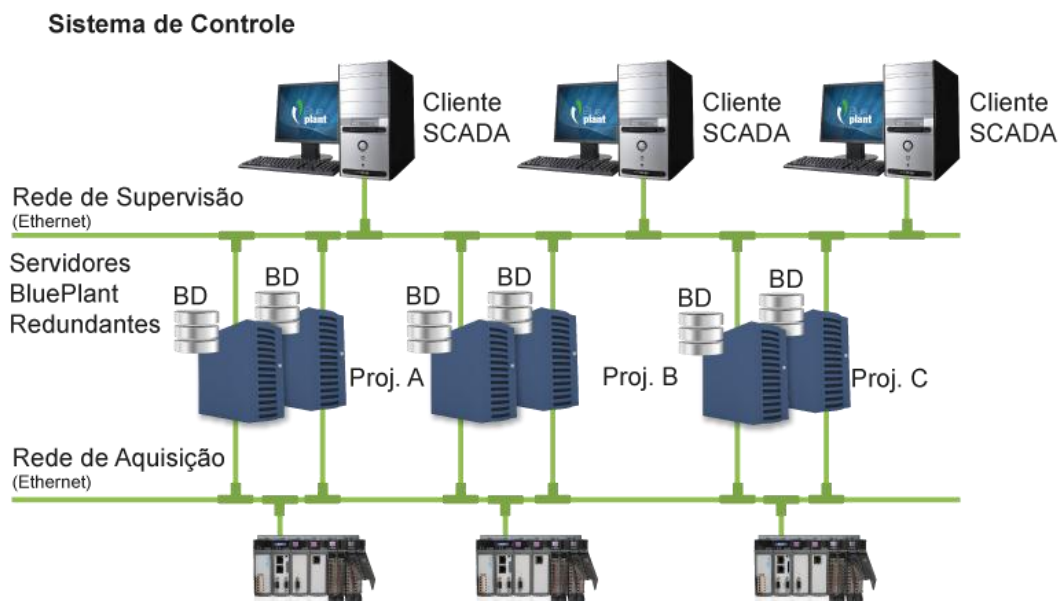



Figura 7-7. Sistema de Controle

Nesse cenário, o sistema é organizado em locais discretos controlados por operadores locais apoiados pelos servidores redundantes locais. Ao mesmo tempo, pode-se caracterizar um nível de gestão em uma sala de controle central que monitora simultaneamente todos os sites. Cada site é representado no projeto como um cluster separado, agrupando seus servidores primário e standby.

Sistema de Controle Distribuído e Distribuido Redundante

Esse sistema contempla uma maquina BluePlant servidora e módulos Alarme, Historiador, Banco de Dados e Clientes SCADA em outros computadores da rede LAN, de maneira que cada módulo é executado em um computador distinto. A Figura 7-8 mostra esse sistema.

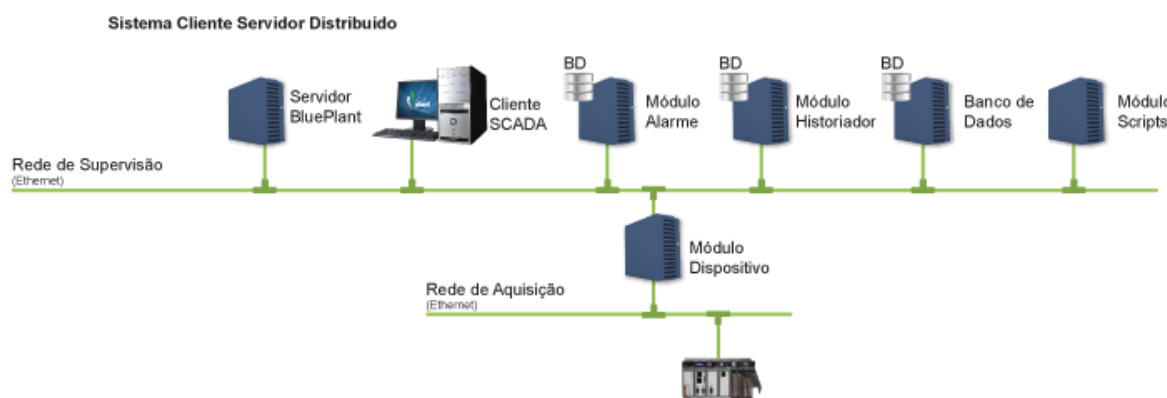



Figura 7-8. Sistema de Controle Distribuído

A arquitetura cliente-servidor permite que os módulos sejam distribuídos através de vários computadores em uma LAN, criando um sistema que oferece benefícios de flexibilidade e desempenho. Cada componente é identificado, no âmbito do projeto, por um endereço, permitindo que sua localização e requisitos de hardware possam ser considerados de forma independente. Este tipo de arquitetura também pode ser configurada como um sistema de cluster único.

O próximo cenário trata-se de um sistema redundante, mas com seu par em localização diferente. Esta configuração demanda uma rede com elevadas taxas de transmissão entre as localidades, visto que a quantidade de dados trocada entre os pares servidores é expressiva. A Figura 7-9 mostra essa configuração.

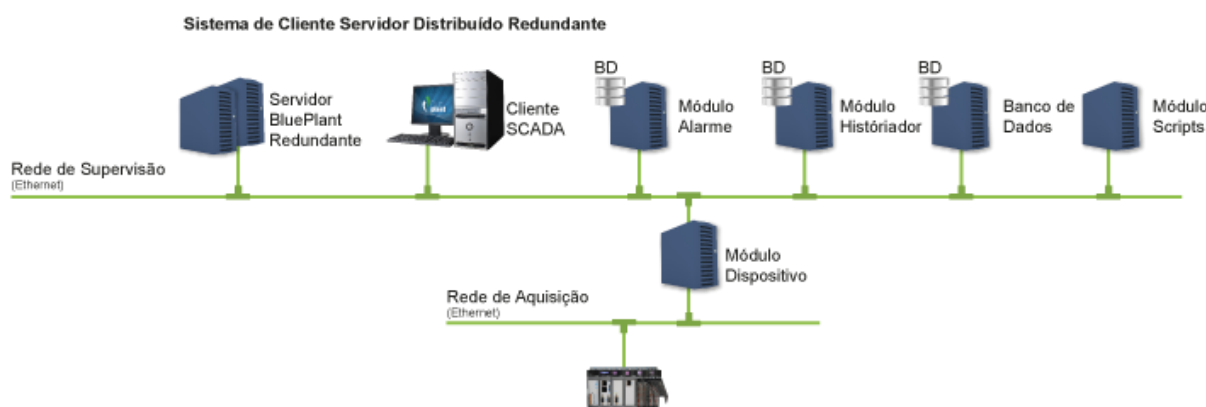



Figura 7-9. Sistema de Controle Distribuído Redundante

Neste cenário, o projeto contempla sites operados localmente cada um contendo seu próprio conjunto de servidores e clientes.

No caso de algum site tornar-se inoperante, o monitoramento ininterrupto é garantido distribuindo-se os servidores primário e secundário entre os diferentes sites, ou colocando-se os servidores como secundário numa localização central.

Um cluster é usado para definir o papel dos servidores em cada local, podendo ser visualizado em um projeto comum, sendo executado em cada cliente.

Sistema de Compartilhamento de Carga

É similar ao sistema cliente e servidor mencionado anteriormente, mas com redundância somente no módulo de dispositivo. Esse sistema possibilita uma melhor utilização da infraestrutura disponível, visto que permite o compartilhamento da carga de trabalho entre diferentes computadores e UCPs.

Esta abordagem pode ser usada para melhorar o desempenho da rede, os tempos de acesso de dados e a estabilidade geral do sistema.

Além disso, por meio de clusters é possível executar múltiplos servidores do mesmo tipo em um computador único.

Por outro lado, a distribuição dos servidores em dois clusters permite que os mesmos operem como unidades redundantes entre si reduzindo o número de computadores exigidos.

8. Glossário

AppDomain	Limite fornecido pelo CLR (Common Language Runtime) englobando os objetos criados a partir do mesmo escopo da aplicação. Os domínios da aplicação ajudam a isolar os objetos criados em uma aplicação de outros objetos criados em outras aplicações. O TRuntime.exe é executado em um AppDomain diferente (módulo do dispositivo ou visualizador) dos outros módulos. Termo relacionado ao Microsoft dotNet.
Assembly	"Assembly" refere-se a um arquivo executável (.EXE) ou biblioteca (.DLL) criada com o uso de código gerenciado e framework Microsoft .NET. Termo relacionado ao Microsoft dotNet.
Designer	Componente de desenvolvimento do gerenciador do programa usado para projetar telas sinóticas. Telas sinóticas são representações gráficas de processos industriais gerados no ambiente de desenho.
Device Module	Módulo que permite a implementação de protocolos de comunicação (Dispositivos) em máquinas remotas com o servidor no qual o projeto está sendo executado, mantendo o Runtime.
Domain, Server Domain, Client Domain	Refere-se aos valores e à localização de objetos quando em execução. "Objeto de domínio de servidor em execução no servidor" significa o objeto em execução no servidor. Valores associados a esse objeto abrangem todo o sistema. "Domínio do cliente" significa que o objeto está sendo executado na estação do cliente e cada máquina cliente pode apresentar valores diferentes.
Internal Module	Características e programas que implementam funções do sistema interno que são executadas sem nenhuma configuração do usuário. Exemplos: tarefa de sincronização de rede e gerador de relatório de plano de fundo. Algumas ferramentas BluePlant, por exemplo, ModuleInformation.exe, podem exibir informações de status destes módulos internos. No entanto, estas informações são exigidas somente para otimizações de sistemas avançados. Exemplo: módulo de sincronização.
Main Project File	"Main Project File" refere-se à base de dados SQL do BluePlant-codificado que contém a configuração do projeto. A extensão .TPROJ mostra o projeto atual em desenvolvimento. A extensão .TENG mostra os projetos Somente Leitura publicados.
Modifier (Device/Points)	Parâmetro auxiliar para marcar pontos de leitura e escrita em um dispositivo, tratamento de arrays, máscaras de bit, strings, swap e outras operações nas quais a definição de Tipo de Dado não pode caracterizá-los completamente (parâmetros adicionais são exigidos para definir o Tipo de Dado).
Module	Programa que acessa o banco de dados em tempo real (RtDB), podendo ser composto de um ou mais Assemblies. Exemplo: Manager.XBAP é um módulo de configuração de projeto que fornece ao usuário o acesso ao seu nome online através do URL do navegador.
Namespace	Espaço de endereço. Todos os objetos criados com Namespace apresentam nomes exclusivos. O Namespace também pode ser entendido como uma configuração para criar hierarquia entre os objetos. Exemplo: todas as variáveis de processo são agrupadas no Tag, namespace, todos os relatórios criados são agrupados no relatório namespace.
ObjectType (RunObj type)	Determina o tipo de objetos de Runtime. Por exemplo, um objeto pode apresentar os seguintes tipos de Runtime: display, report e ScriptFunction,Tela, Relatório ou Função Script.
Objects, RunTime Objects	Objetos de Runtime são os objetos visíveis (através dos seus nomes) acessados via scripts ou exibidos no projeto. Exemplos: tags, Relatórios, Telas e outros. Também deve ser usado pelos desenvolvedores de módulo. Um módulo de alarme, por exemplo, inicializará e atualizará os valores dos objetos de Runtime associados com os alarmes.
ObjectValues, PropertyValue	Objetos de Runtime (tags, Nós de dispositivos, etc.) podem ter uma ou mais propriedades definidas. O objeto TAG (e seu valor) apresenta propriedades, tais como: mínimo, máximo e descrição. Durante a configuração específica do projeto são exigidos os itens "PropertyValue" ou "ObjectValue". Especifique o nome completo até o valor da propriedade final, isto é, TAG.myTag.Min ou Tag.MyTag.Description. Para algumas configurações como itens de alarme, historiador ou pontos do dispositivo especificando o OBJETO principal (Tag.MyTag neste exemplo), o sistema assumirá, por padrão, a propriedade de valor para executar a operação.
ObjectWatch	Utilidade de diagnóstico para verificar e modificar valores dos objetos quando em execução.
Parameters (TagConfiguration)	Definição dos parâmetros de comportamento e valores dos tags de processamento. Exemplo: CounterUP para contadores, ou DeadBand smoothing tags para Inteiros.
Project	Conjunto de configurações BluePlant, desenhos de telas, relatórios, documentos e notas do usuário criadas e editadas como uma entidade única.
projectDB or ProjectDatabase	ProjectDB (or ProjectDatabase) é uma base de dados proprietária com informações de configuração. Corresponde aos arquivos com extensão <projeto>.Tproj (por exemplo, Atual) ou <projeto> _ VERSION.Teng (tipo Release).
Property	Propriedade (valor) associado a um Tag ou a um objeto de Runtime.
RunDB, Runtime Database	Base de dados de tempo real criada quando o projeto está em execução. Todos os objetos de Runtime acessáveis via script (tais como tags e relatórios) são gerados internamente com código encapsulado e

	acessáveis através da classe RunDB. O RunDB corresponde ao primeiro nível da árvore de objetos de Runtime.
Runtime	Projeto com módulos de Runtime carregados e em execução.
Runtime Startup	Operação que permite a execução de um projeto. Esta operação pode ser executada a partir do programa TStartup.exe na versão atual do BluePlant, ou a partir do TServer.exe na versão publicada do projeto.
RuntimeDB	RuntimeDB é uma cópia do banco de dados do ProjectDB que contém informações específicas sobre o RuntimeDB quando a versão do Runtime é publicada. Corresponde aos arquivos com <projeto> _ <versão>.trun.
Tag	Variável de processo. Nome de um Namespace que inclui todas as variáveis criadas pelo usuário em uma configuração do projeto.
Tag type	Define o tipo de objetos no Tag do Namespace. Exemplo: Digital, Analógico e Texto. Esses tags representam uma classe de propriedades acessadas diretamente, tais como: Min, Max, valor, Qualidade. Cada propriedade é criada internamente como ValueType.
Task (Script.Task)	Programa de tarefas escrito em VB.NET (ou C #) executado no servidor ou no cliente durante o Runtime do projeto. A execução será no servidor ou cliente dependendo da propriedade de domínio configurada no script.
TManager	Programa que realiza a configuração do projeto.
Toggle	Inverte o valor de uma variável. Valores maiores que zero são convertidos a zero; zero é convertido ao valor "1".
Partial Trust, Full Trust	Partial Trust: Ambiente no qual uma aplicação tem acesso limitado aos recursos (acesso restrito a arquivos de pastas particulares, execução de outras aplicações, etc.). Aplicações XBAP que são executadas com um navegador (Internet Explorer, por exemplo) devem utilizar o modo "Partial Trust". Full Trust: Ambiente no qual uma aplicação tem acesso a todos os recursos do sistema. Aplicações instaladas em um computador geralmente são executadas no modo "Full Trust". Termo relacionado ao Microsoft dotNet.
TWelcome	BluePlant inicial e tela de seleção do projeto.
Visibility (Tag Visibility)	Refere-se ao sistema de Tag. Os tags podem ser públicos, protegidos ou privados. Público: o valor do Tag durante a execução está disponível para acesso por programas externos via TCP/IP ou Servidor OPC. O valor do Tag é necessariamente global ou compartilhado em todas as estações cliente (Domínio do Servidor). Protegido: somente leitura. Privado: um Tag definido como "Privado" não pode ser acessado por aplicações externas (Servidor OPC, TCP/IP) e será executado em um Cliente ou em um Servidor, de acordo com a configuração da aplicação com as seguintes características: • Tags usados somente em módulos chamados pelo cliente (tais como telas e Relatórios) e que são executados no escopo deste. Podem apresentar valores diferentes em cada máquina cliente (Domínio do Cliente); • Tags usados em modulo servidores, tais como dispositivos e alarmes. Apresentam um único valor no sistema (Domínio do Servidor).
Xbap	XAML browser application (XBAP) é uma aplicação executada dentro de um browser (IE). Termo relacionado ao Microsoft dotNet.